ADVANCES IN PHASE RETRIEVAL AND BLIND DECONVOLUTION

A thesis presented for the degree of
Doctor of Philosophy
in Electrical and Electronic Engineering,
at the
University of Canterbury,
Christchurch, New Zealand.

by
Bruce C. McCallum

May 1990

Abstract

A number of new techniques for solving multi-dimensional phase problems and blind deconvolution problems are presented. Some of these techniques augment existing algorithms, while others constitute original algorithms.

The phase problem and blind deconvolution problem are reviewed. A comprehensive description of existing methods for effecting phase retrieval and blind deconvolution is presented.

The behaviour of Fienup's hybrid input-output phase retrieval algorithm, when it operates on appreciably contaminated data, is demonstrated. The image error, which expresses the violation of the image-space constraints, is shown to exhibit pronounced fluctuations. It is demonstrated that significantly improved final image-forms are obtained by appropriately averaging the image-forms generated at those iterations for which the image error has locally minimum value.

A technique for reducing the computational requirements of Fienup's hybrid input-output algorithm is introduced and demonstrated. This technique involves initially estimating the phases of low spatial frequency components of the image, by operating on a fraction of the total number of available Fourier magnitude samples. The number of magnitude samples on which the algorithm operates is increased as the iterations proceed, until all of the available samples are employed.

An alternative input-output algorithm is presented, which comprises an appropriate combination of Fienup's basic and hybrid input-output algorithms. It is demonstrated that the performance of this algorithm is superior to that of the hybrid input-output algorithm, in situations where positivity is the only constraint that can be enforced in image space.

A phase retrieval algorithm is proposed which employs simulated annealing. Unlike the existing simulated annealing phase retrieval algorithm, in which samples of the image are perturbed, this new algorithm operates by perturbing the phases of the Nyquist samples in Fourier space.

A new blind deconvolution algorithm, which invokes simulated annealing, is presented. It is demonstrated that this algorithm is capable of effecting the blind deconvolution of contaminated convolution data.

An algorithm is presented and illustrated for directly (i.e. non-iteratively) blindly deconvolving the convolution of two images, where each image comprises a collection of points. A related algorithm is introduced, specifically for effecting the blind deconvolution of ghosted shift-and-add images of star clusters. The performance of this algorithm, when operating on shift-and-add images computed from simulated speckle data, is demonstrated.

Acknowledgements

Many people have, in various ways, assisted with my Ph.D. research and with the preparation of this thesis. But there are some who deserve particular mention.

I am grateful to my supervisor, Professor Richard Bates, for his continual guidance and encouragement. My colleagues Richard Fright, Richard Lane, Mike Hayes, Peter Gardenier, and Bruce Davey deserve special thanks, for the many valuable discussions that I had with them, and for the various software that they provided. I am indebted to Mike Hayes and Jiang Hong for their proof-reading of this thesis.

Financial assistance from the New Zealand University Grants Committee and the Division of Information Technology of the Department of Scientific and Industrial Research is gratefully acknowledged.

Finally, I would like to thank my family, friends, and flatmates for the support, friendship, and understanding that they have offered over the last few years.

Contents

Preface

This thesis presents a number of new results concerning various aspects of two image reconstruction problems, namely the phase problem and the blind deconvolution problem. The purpose of this preface is to describe how these somewhat disparate studies came about.

In 1985, I entered the Electrical and Electronic Engineering Department of the University of Canterbury as a research student. Being interested in the general area of signal processing, I was privileged to join Professor R. H. T. Bates' research group. This group is renowned for its work, spanning the last two decades, in a number of diverse areas of scientific endeavour. Such areas include radio engineering, astronomical imaging, inverse problems (especially computed tomography), and biomedical engineering. Members of Professor Bates' research group who, at the time of my entry into the group, were researching topics that relate to those in this thesis include Richard Fright (1984), Alastair Sinton (1986), and Daniel Mnyama (1987). Contemporaries of mine who performed research related to that presented here include Richard Lane (1988), Bruce Davey (1989), and Peter Gardenier (1990).

In collaboration with Peter Gardenier, the first significant research project that I tackled was related to pattern recognition. We implemented an algorithm devised by Professor Bates which, when presented with an image of some object, generates a descriptor that is invariant to the position, orientation, size and brightness of the object. An advantage of this particular descriptor is that the object is (in general) uniquely recoverable from the descriptor. One of the steps in this algorithm is to retain the magnitude of the Fourier transform of the input image. Recovery of the object from the descriptor thus poses the problem of inferring an image from the magnitude of its Fourier transform, which is known as the phase problem. Although this work on pattern recognition is not mentioned further in this thesis, it has been reported by Gardenier *et al.* (1986a) and McCallum *et al.* (1986).

After I had gained some appreciation of the phase problem from this project, Professor Bates suggested that it would be worthwhile to investigate some aspects of the performance of Fienup's (1982) hybrid input-output phase retrieval algorithm. It was fortunate that Richard Fright was a Postdoctoral Fellow in the Electrical and Electronic Engineering Department at this time, since he had gained considerable experience in this area while performing the research reported in his doctoral thesis. On studying the performance of the hybrid input-output phase retrieval algorithm when it operates on appreciably contaminated data, we noticed that the behaviour of this algorithm was rather erratic. This led us to the strategies that are described in Chapter 4.

On becoming acquainted with the hybrid input-output algorithm, it was evi-

dent that any reduction to the amount of computation required, to recover an image (especially one containing many pixels) from its Fourier transform magnitude, would be valuable. It seemed to us that it should be possible to initially apply the hybrid input-output algorithm to only a fraction of the available number of Fourier magnitude samples, thereby recovering the phases of the low spatial frequency components. The algorithm which resulted is described in §5.2.

After studying the basis of the hybrid input-output algorithm, it became clear that its derivation was somewhat *ad hoc*. This suggested that it may be worthwhile to investigate other input-output algorithms. A few algorithms were devised and tested, one of which is described in §5.3.

While pursuing the research into the phase problem described above, several of my colleagues were working on various aspects of another image reconstruction problem, namely the blind deconvolution problem. This problem involves attempting to recover the components of a convolution, from knowledge of the convolution alone. It was the advent of the zero-sheet concept (Lane and Bates, 1987a) which showed that all multi-dimensional blind deconvolution problems are potentially solvable, under ideal circumstances. Lane and Bates (1987b), followed by Davey *et al.* (1989), then demonstrated that blind deconvolution problems can also be solved in practical situations, when the convolution data are contaminated. Because the phase problem is a special case of the blind deconvolution problem, we began to contemplate whether any existing phase retrieval techniques could be usefully generalized in order to apply to the blind deconvolution problem.

One such technique which seemed suitable for application to the blind deconvolution problem was simulated annealing. This led to the development of the algorithm that is described in Chapter 6.

While working on applying simulated annealing to the blind deconvolution problem, it occurred to us that there was another way in which simulated annealing could be used to solve the phase problem. This involved employing simulated annealing in Fourier space, rather than in image space. The resulting algorithm is described in §5.4.

A blind deconvolution problem that arises in optical astronomy, is the deghosting of images produced by the shift-and-add algorithm. A ghosted shift-and-add image of a star cluster typically contains a number of shifted and scaled versions of the original cluster. It seemed to us that it should be possible to blindly deconvolve such an image, by an appropriate modification of the technique devised by Fienup (1981a) to recover an image of a star cluster from its autocorrelation. The blind deconvolution algorithms which resulted are described in Chapter 7.

The preceding narrative describes how the work presented in this thesis came about. It is now appropriate to give a chapter by chapter outline of the thesis.

Relevant background material is presented in Chapters 1 to 3. Chapter 1 introduces the terminology, notation and mathematical tools that are used in subsequent chapters.

In Chapter 2, the phase problem is introduced, and some important aspects of the phase problem are discussed. These include the dominance of spectral phase over spectral magnitude in determining the form of an image; the manner in which spectral phase is constrained by spectral magnitude; and the question of uniqueness

of solutions to phase problems. A number of examples of practical phase problems are given. The blind deconvolution problem is also introduced, and the question of uniqueness of solutions is discussed. Finally, some practical blind deconvolution problems are described.

Chapter 3 begins by presenting various techniques for solving general image reconstruction problems. Detailed descriptions of a number of techniques for solving the phase problem are then given. Finally, several techniques for solving the blind deconvolution problem are described.

All of the work described in Chapters 4 to 7 constitutes original research. The software to implement each of the algorithms that are illustrated in Chapters 4 to 7 was written by myself. This software interfaces to the IMPROC image processing package, originally written by R. G. Lane.

Chapter 4 demonstrates the erratic behaviour of the hybrid input-output algorithm when it operates on appreciably contaminated data, and discusses the resulting difficulty in attempting to obtain a faithful reconstruction in such situations. An algorithm which averages a number of appropriately selected reconstructed images, in order to produce an enhanced image, is described and demonstrated.

Three strategies for reducing the computational requirements of phase retrieval algorithms are described in Chapter 5. A technique for accelerating the convergence of the hybrid input-output algorithm, by operating upon an increasing number of Fourier magnitude samples, is detailed in §5.2. An alternative to the hybrid input-output algorithm, which comprises a combination of the basic and hybrid input-output algorithms, is presented in §5.3. Finally, §5.4 describes a phase retrieval algorithm that employs simulated annealing, which operates by appropriately varying the spectral phases of the image.

Chapter 6 describes an algorithm to effect blind deconvolution by simulated annealing. Results of the algorithm's operation on contaminated convolution data are given.

Two algorithms are presented and demonstrated in Chapter 7, which can blindly deconvolve the convolution of two images, in situations where each image comprises a collection of points. One algorithm is devised specifically for effecting the blind deconvolution of ghosted shift-and-add images, while the other is capable of blindly deconvolving a general convolution.

Chapter 8 is the final chapter of this thesis, where conclusions are drawn, and a number of suggestions for further research are made.

Publications and presentations that were prepared during the course of this research are now listed:

GARDENIER, P.H., MCCALLUM, B.C. and BATES, R.H.T. (1986a), 'Fourier transform magnitudes are unique pattern recognition templates', *Biological Cybernetics*, Vol. 54, pp. 385–391.

MCCALLUM, B.C., GARDENIER, P.H. and BATES, R.H.T. (1986), 'Invertible invariant transformations for robotic catalogues', in *Proceedings of the International Conference on Future Computing Systems*, pp. 151–158.

MCCALLUM, B.C. and BATES, R.H.T. (1989), 'Towards a strategy for auto-

matic phase retrieval from noisy Fourier intensities', *Journal of Modern Optics*, Vol. 36, No. 5, May, pp. 619–648.

MCCALLUM, B.C. (1990), 'Blind deconvolution by simulated annealing', *Optics Communications*, Vol. 75, No. 2, pp. 101–105.

Glossary

Most of the conventions, symbols, and abbreviations that are employed throughout this thesis are defined in that part of the text at which they are first invoked. Nevertheless, they are repeated here for ease of reference.

Conventions

Quantities of interest in this thesis exist in K-dimensional spaces known as image-space, Fourier-space, and z-space, and are referred to as images, spectra, and z-spectra respectively (see §§1.1, 1.3 and 1.7). The position vectors $\vec{x}$, $\vec{u}$, and $\vec{\zeta}$ respectively specify arbitrary points in each of these spaces. Lowercase letters are used to identify images, the spectra and z-spectra of which are identified by the corresponding uppercase and uppercase script letters respectively. For example, the spectrum and z-spectrum of the image $q(\vec{x})$ are denoted by $Q(\vec{u})$ and $\mathcal{Q}(\vec{\zeta})$ respectively. Position vectors in discrete image space and discrete Fourier space are respectively denoted by $\vec{m}$ and $\vec{p}$. Thus m_k, the k^{th} component of $\vec{m}$, can only assume integer values. Functions of continuous variables are indicated by enclosing their arguments in round brackets, e.g. $q(\vec{x})$, whereas functions of discrete variables are indicated by enclosing their arguments in square brackets, e.g. $q[\vec{m}]$. A function with an argument which is itself a function is indicated by enclosing the argument in curly brackets, e.g. $F\{q(\vec{x})\}$. For $K = 1$ and $K = 2$, it is convenient to invoke the notations listed in the table below.

	continuous argument			discrete argument		
space	$K=1$	$K=2$	K arbitrary	$K=1$	$K=2$	K arbitrary
image	$f(x)$	$f(x,y)$	$f(\vec{x})$	$f[m]$	$f[m,n]$	$f[\vec{m}]$
Fourier	$F(u)$	$F(u,v)$	$F(\vec{u})$	$F[p]$	$F[p,q]$	$F[\vec{p}]$
z	$\mathcal{F}(\zeta)$	$\mathcal{F}(\zeta,\gamma)$	$\mathcal{F}(\vec{\zeta})$	–	–	–

Symbols

i	the pure imaginary, where $i^2 = -1$
$\odot$	the convolution operator (§1.4)
$\odot^{-1}$	the deconvolution operator, where $a(\vec{x}) \odot^{-1} b(\vec{x})$ indicates the deconvolution of $b(\vec{x})$ from $a(\vec{x})$

$*$	the correlation operator (§1.4)
$\{q_m : m = 1, \ldots, M\}$	a set of M elements, the m^{th} of which is q_m
I^+	the set of positive integers
$\in$	is an element of
$\notin$	is not an element of
$\bigcup$	set union operator
$\bigcap$	set intersection operator
$\forall$	for all
$\delta(x)$	the delta or impulse function (§1.2)
$\text{rect}(x)$	the rectangle function (§1.2)
$\text{tri}(x)$	the triangle function (§1.2)
$\text{sinc}(x)$	the sinc function (§1.2)
$< q_m >$	the ensemble average of the set $\{q_m\}$, given by $< q_m > = \frac{1}{M} \sum_{m=1}^{M} q_m$
$\overline{q(t)}$	the time average of $q(t)$, given by $\overline{q(t)} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} q(t)\, dt$
$\|\vec{x}\|$	the norm of $\vec{x}$ (§3.1.1.2)
mod	the modulo operator
min	the minimum value of
$q^{M \times N}[m, n]$	explicitly indicates that $q[m, n]$ comprises $M \times N$ pixels (§5.2.2)
$\hat{q}(\vec{x})$	an estimate of $q(\vec{x})$
$q^*(\vec{x})$	the complex conjugate of $q(\vec{x})$ (§1.2)
$qq(\vec{x})$	the autocorrelation of $q(\vec{x})$ (§1.4)
$\|q(\vec{x})\|$	the magnitude of $q(\vec{x})$ (§1.2)
$\text{Re}\{q(\vec{x})\}$	the real part of $q(\vec{x})$ (§1.2)
$\text{Im}\{q(\vec{x})\}$	the imaginary part of $q(\vec{x})$ (§1.2)
$\text{Ph}\{q(\vec{x})\}$	the phase of $q(\vec{x})$ (§1.2)
$\text{En}\{q[\vec{m}]\}$	the energy of $q[\vec{m}]$ (§1.2)
$\text{rms}\{q[\vec{m}]\}$	the rms value of $q[\vec{m}]$ (§1.2)
$\mathcal{Z}\{\mathcal{F}(\vec{\zeta})\}$	denotes the zero-sheet of $\mathcal{F}(\vec{\zeta})$ (§2.4.2) if $K > 1$, otherwise denotes the set of zeros of $\mathcal{F}(\vec{\zeta})$ (§1.7)
$S_q(\vec{x})$	the support of $q(\vec{x})$ (§1.5)
$B_q(\vec{x})$	the image box of $q(\vec{x})$ (§1.5)
L_{q,x_k}	the extent of $q(\vec{x})$ in the x_k direction (§1.5)
snr	the signal to noise ratio (§1.2)
cl	the contamination level (§1.2)
$\longleftrightarrow$	connects a Fourier transform pair (§1.3)
$\text{FT}\{q(\vec{x})\}$	the Fourier transform of $q(\vec{x})$ (§1.3)
$\text{FT}^{-1}\{Q(\vec{u})\}$	the inverse Fourier transform of $Q(\vec{u})$ (§1.3)
$\text{FFT}\{q[\vec{m}]\}$	the FFT of $q[\vec{m}]$ (§1.6)
$\text{IFFT}\{Q[\vec{p}]\}$	the IFFT of $Q[\vec{p}]$ (§1.6)
$\rightarrow$	tends to
$o(v)$	of the order of v
Å	Angstrom (10^{-10}m)
arc-sec	arc-second ($\frac{1}{3600}^\circ$)

Abbreviations

DFT	discrete Fourier transform
FFT	fast Fourier transform
IFFT	inverse fast Fourier transform
LHS	left hand side
psf	point spread function
RHS	right hand side
rms	root-mean-square

Chapter 1

Preliminaries

The aim of this chapter is to introduce the terminology, notation, and mathematical tools that are used in subsequent chapters of this thesis. A decimal system of section numbering is adopted, and sections are referred to in the text by the symbol § followed by the section number. Thus §2.3.1 refers to the first subsection of the third section of chapter 2. Equations are numbered consecutively within each chapter, and referred to by enclosing the equation number in brackets. Figures and algorithms are numbered in the same manner as equations, and are referred to by preceding the figure or algorithm number with "Figure" or "Algorithm".

A typical image restoration scenario is described in §1.1, which encompasses many of the practical applications discussed in later chapters. Notation for useful mathematical quantities such as vectors and complex numbers is given, and some useful functions are defined, in §1.2. The Fourier transform is fundamental to the work described in this thesis, and is introduced in §1.3, where its importance is explained. Convolutions and related operations, and useful characteristics of them, are described in §1.4. The inevitability of having to sample data before performing any computation is mentioned in §1.5, and the importance of sample spacing is discussed. When handling sampled data, it is appropriate to use the discrete Fourier transform (DFT), which is described in §1.6. The implications of approximating the continuous Fourier transform by the DFT are also discussed, and the fast Fourier transform, a computationally efficient implementation of the DFT, is described. This chapter concludes with a description of the z-transform, and a brief introduction to complex zeros.

1.1 Objects and Images

In numerous seemingly disparate branches of science and engineering, situations arise where one wishes to obtain information about some physical entity, which is here called the *object*. To further this aim, one collects a set of *measurements*. From this set of measurements, one attempts to form an estimate of the object, here called the *image*. However, this final step, known as the *reconstruction process*, typically requires some *a priori* information concerning the relationship between the object and the measurements. Further *a priori* information concerning the object itself is often also required. This whole process is illustrated diagrammatically in Figure 1.1.

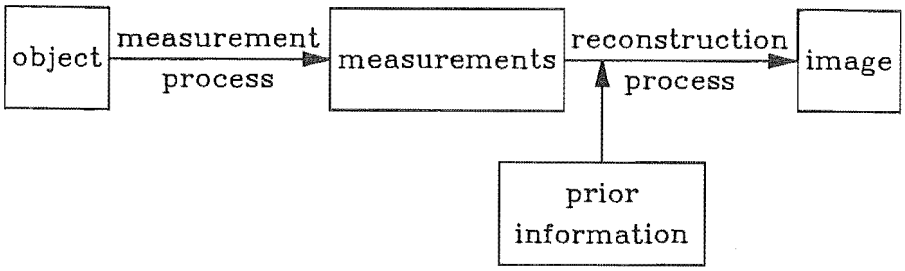


Figure 1.1 The image reconstruction scenario.

The reconstruction process tends to be further complicated by a number of factors, the more common of which are now mentioned. Measurements can only be made over a finite region of the space in which the measured quantity exists. This is because the number of measurements must always be finite in practice, being limited by differing factors depending on the particular application. If the measurements are made consecutively, the effort involved in making each measurement, or the time taken to do so, may limit the number of measurements. If all measurements are made at once, with an instrument with multiple sensors, the number of measurements is of course limited by the number of sensors. Sometimes, the limiting factor is not the number of measurements that can be made, but rather the number that can be handled once obtained. Such limitations may occur in either the transmission, storage, or processing of the measurements.

Every real-world measurement inevitably contains *contamination* of some sort, being the difference between the actual measurement obtained and that which would have been obtained under ideal circumstances. Contamination may result from the quantity being measured differing from its ideal value, due for example to noise on, or disturbances to, the measured quantity. Alternatively, contamination may result from a measurement differing from the ideal measured value, due to such factors as transducer (and/or recorder) offsets and nonlinearities.

Measurements may be *incomplete*, which is here taken to mean that the measurements are insufficient to fully characterize the object without additional *a priori* information concerning the object, or alternatively that the image cannot be obtained from the measurements by a simple-minded inversion strategy. Measurements of either only the magnitude or only the phase of a complex quantity are common examples of incomplete measurements.

In many situations, the direct measurement of the object is precluded by practical limitations, but it may instead be possible to make measurements of the radiation emitted (or scattered) by the object. This is commonly referred to as *remote sensing* or *remote probing*. An example of such a situation is x-ray crystallography, in which the minute size of the object (the unit cell of a crystal) precludes direct measurements being made. However, much useful information can be obtained by illuminating the crystal with x-rays and measuring the scattered field. In such instances, the space in which the image (and, equivalently, the object) exists is called *image space*, whilst the space in which the measurements exist is here called *measurement space*. It so happens that in numerous physical situations, the measurements are of, or closely

related to, the Fourier transform of the object (see §1.3), in which case measurement space is known as *Fourier space.*

1.2 Notation and Conventions

Throughout this thesis, a function of a continuous variable (or variables) is indicated by enclosing the variable(s) in round brackets, whereas a function of a discrete variable is indicated by enclosing the variable(s) in square brackets. For example, $g(x)$ is a function of the continuous variable x, but $g[m]$ is a function of the discrete variable m, which can only take integer values. Furthermore, a function of a function (or of an image) is indicated by enclosing the argument in curly braces. For example, the function H acting on the image $f(x,y)$ is written as $H\{f(x,y)\}$. Note that in such cases, the functional dependence of the argument is often omitted, for conciseness, if no confusion is caused by doing so. Thus $H\{f(x,y)\}$ may simply be written as $H\{f\}$.

Vector quantities are indicated by adorning the variable with an arrow, such as $\vec{x}$, the k^{th} component of which is x_k. Position vectors in K-dimensional image and Fourier space are written as $\vec{x}$ and $\vec{u}$ respectively. A quantity in image space is denoted by a lower case letter, and its corresponding quantity in Fourier space is denoted by the corresponding upper case letter, e.g. the Fourier transform of an image $f(\vec{x})$ is $F(\vec{u})$. For the majority of applications discussed in this thesis, $K = 2$, in which case the notations $f(x,y)$ and $F(u,v)$ are invoked for $f(\vec{x})$ and $F(\vec{u})$ respectively. For $K = 1$, the notations $f(x)$ and $F(u)$ are used for $f(\vec{x})$ and $F(\vec{u})$ respectively.

The following notations apply to complex quantities, typically c, whose *real* and *imaginary parts* are written as $\mathrm{Re}(c)$ and $\mathrm{Im}(c)$ respectively. Thus

$$c = \mathrm{Re}(c) + i\,\mathrm{Im}(c) \tag{1.1}$$

where $i = \sqrt{-1}$, the pure imaginary. The *magnitude* (or *modulus*) of c is denoted by $|c|$, and the *phase* (or *argument*) of c by $\mathrm{Ph}(c)$, thus

$$|c| = \sqrt{\mathrm{Re}(c)^2 + \mathrm{Im}(c)^2} \tag{1.2}$$

and

$$\mathrm{Ph}(c) = \tan^{-1}\frac{\mathrm{Im}(c)}{\mathrm{Re}(c)} \tag{1.3}$$

Note that

$$c = |c|e^{i\,\mathrm{Ph}(c)} \tag{1.4}$$

The square of the magnitude of c is called the *intensity* of c. The *complex conjugate* of c is denoted by c^*, where

$$c^* = \mathrm{Re}(c) - i\,\mathrm{Im}(c) = |c|\,e^{-i\,\mathrm{Ph}(c)} \tag{1.5}$$

Four useful functions are now defined for use in subsequent sections. Dirac's *delta function* (also known as the *impulse function*) is defined (*cf.* Bates and McDonnell, 1986, §6) by

$$\int_{x\in O} \delta(x)\,dx = 1 \quad \text{and} \tag{1.6a}$$

$$\int_{x\notin O} \delta(x)\,dx = 0 \tag{1.6b}$$

where O is an infinitesimally small interval of the x-axis containing $x = 0$.

The *rectangle function* is defined (Bracewell, 1978, ch. 4) as

$$\text{rect}(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

The *triangle function* is defined (Bracewell, 1978, ch. 4) as

$$\text{tri}(x) = \begin{cases} 1 - |x| & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

The *sinc function* is defined (Bracewell, 1978, ch. 4) as

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{1.9}$$

Four useful definitions are now introduced, which are conveniently stated in terms of the discrete one-dimensional function $f[m]$. Their generalizations to higher dimensions are obvious.

The *energy* of $f[m]$, denoted by $\text{En}\{f[m]\}$, is defined as

$$\text{En}\{f[m]\} = \sum_{m=-\infty}^{\infty} |f[m]|^2 \tag{1.10}$$

The *root-mean-square* or *rms value* of $f[m]$, which is denoted by $\text{rms}\{f[m]\}$, is defined as

$$\text{rms}\{f[m]\} = \sqrt{\frac{1}{M} \sum_{m} |f[m]|^2} \tag{1.11}$$

where M is the number of values over which the summation is performed.

Consider a situation in which $f[m]$ is degraded by additive contamination $c[m]$, thereby giving rise to $f_c[m]$, where $f_c[m] = f[m] + c[m]$. The *contamination level* of $f_c[m]$, denoted by cl, is defined by

$$cl = \text{En}\{c[m]\} \,/\text{En}\{f[m]\} \tag{1.12}$$

It is sometimes more convenient to discuss the *signal-to-noise ratio* of $f_c[m]$, denoted by snr, which is simply given by

$$snr = \frac{1}{cl} \tag{1.13}$$

1.3 The Fourier Transform

The Fourier transform $F(\vec{u})$ of an image $f(\vec{x})$ is defined (Bates and McDonnell, 1986, §6) as

$$F(\vec{u}) = \int_{-\infty}^{\infty} f(\vec{x}) e^{i2\pi \vec{u} \cdot \vec{x}} d\vec{x} \tag{1.14}$$

where $d\vec{x}$ is the K-dimensional volume element, and $\vec{u} \cdot \vec{x}$ denotes the scalar product of $\vec{u}$ with $\vec{x}$, namely

$$\vec{u} \cdot \vec{x} = \sum_{k=1}^{K} u_k \, x_k \tag{1.15}$$

The inverse Fourier transform is defined (Bates and McDonnell, 1986, §6) by

$$f(\vec{x}) = \int_{-\infty}^{\infty} F(\vec{u}) e^{-i2\pi\vec{u}\cdot\vec{x}} d\vec{u} \tag{1.16}$$

Note that some authors (Bracewell, 1978, ch. 5) instead define the forward transform with the minus sign, and the inverse transform without. This is of little consequence, so long as one maintains consistency.

One can think of u as representing *spatial frequencies*, and so $F(\vec{u})$ is referred to as the *spatial frequency spectrum*, or simply the *spectrum*, of $f(\vec{x})$. In some situations, as mentioned in §1.1, it is only possible to obtain information relating to $F(\vec{u})$ rather than $f(\vec{x})$. In such circumstances, it is $F(\vec{u})$ that is visible, which has given rise to the term *visibility* as an alternative to spectrum. Equations (1.14) and (1.16) indicate the invertible relationship between an image and its spectrum. It is said that $f(\vec{x})$ and $F(\vec{u})$ constitute a *Fourier transform pair*, a connection which is denoted by

$$f(\vec{x}) \longleftrightarrow F(\vec{u}) \tag{1.17}$$

The operation of performing a Fourier transform or inverse Fourier transform is indicated by the symbols FT and FT^{-1} respectively, thus

$$\mathrm{FT}\{f(\vec{x})\} = F(\vec{u}) \quad \text{and} \quad \mathrm{FT}^{-1}\{F(\vec{u})\} = f(\vec{x}) \tag{1.18}$$

One may wonder why the Fourier transform appears so frequently in image science, and indeed in many other fields. The major reason is that one often encounters a situation where a source of wave motion produces a field that is observed some distance away from the source, in which case (if certain conditions are met) the source and observed fields can be related by the Fourier transform, in the manner outlined below. Examples of some such situations are given in §2.5. A rather more detailed derivation can be found in Goodman (1968, ch. 3,4) or Silver (1965, §6.2).

Consider a finite region in free space containing sources of wave motion, called the *source region*. Cartesian coordinates are set up such that the origin lies within the source region, and the wave motion is of interest in planes parallel to the x, y plane (termed *transverse* planes). Wave motion radiating from the source region can be characterized by a scalar function called the *wave function*, denoted by $\Psi(x, y, z, t)$. The wave motion is observed in the *observation plane*, defined by $z = z_o$, an arbitrary point on which is denoted by the coordinates (x_o, y_o). The wave function $\Psi_o(x_o, y_o, t)$ on the observation plane is called the *observed field* or *diffracted field*. In many practical applications, imaging instruments are placed on the observation plane, in which case it is often referred to as the *aperture plane* or *pupil plane*. The *equivalent source plane* is defined by $z = z_e$, and is chosen so as to lie between the source region and the observation plane. The wave function on the equivalent source plane, denoted by $\Psi_e(x_e, y_e, t)$, is called the *equivalent source field*. The equivalent source plane is so named because one can define a two-dimensional *equivalent source*, lying on the equivalent source plane, which results in an identical observed field as that due to the actual source. If the source is *spatially coherent* (meaning that the radiation from all points of the source is synchronized) and *monochromatic* (meaning that all points of

the source radiate with identical frequency, namely f) then the functional dependence of Ψ can be written as

$$\Psi(x,y,z,t) = \Psi(x,y,z)\, e^{i2\pi ft} \tag{1.19}$$

This harmonic time dependence of Ψ is hereafter omitted, but is taken as understood. The observation plane is said to be in the *far field* if its separation R from the equivalent source plane (i.e $R = z_o - z_e$) satisfies (*cf.* Goodman, 1968, §4.1)

$$R > \frac{2D^2}{\lambda} \tag{1.20}$$

where D is the largest transverse dimension of the source, and λ is the wavelength of the radiation. It transpires (Goodman, 1968, §4.1) that if the observation plane is in the far field, the observed field is related to the equivalent source field by

$$\Psi_o(x_o,y_o) \approx C(R,\lambda,x_o,y_o) \iint \Psi_e(x_e,y_e)\, e^{i2\pi(x_o x_e + y_o y_e)/\lambda R} dx_e\, dy_e \tag{1.21}$$

where C is a complex scale factor with the indicated functional dependence. The significance of (1.21) is that, apart from a scale factor and a multiplicative constant, the observed field and the equivalent source field constitute a Fourier transform pair.

If the source is *spatially incoherent* (meaning that radiations from separated points of the source are statistically independent) then it is the intensity distribution of the wave function that is of interest. Sources of thermally induced radiation (such as those encountered in astronomical applications) are spatially incoherent sources. In such instances, the image is also an intensity distribution, and thus is known to be *positive* (i.e real and non-negative), the significance of which becomes apparent in the following chapter.

A further reason why the Fourier transform has such widespread application is that a problem posed in image space can often take on a different light (and hopefully be simplified) when it is transformed to Fourier space. The deconvolution problem (Bates and McDonnell, 1986, §14) is an example of such a problem, since the process of deconvolution in image space can be transformed into one of factorization in Fourier space. Finally, the advent of an efficient and easily implemented algorithm for computing a discrete approximation to the Fourier transform, namely the fast Fourier transform algorithm (see §1.6), has revolutionized the practical applicability of Fourier transform techniques.

1.4 Convolution and Correlation

A linear system (i.e. one in which superposition applies) can be defined in terms of its *impulse response* $h(\vec{x},\vec{x}')$, which is the output of the system caused by an input consisting of a delta function at $\vec{x}'$ (Papoulis, 1980, §6.5). In image science contexts, $h(\vec{x},\vec{x}')$ describes the image produced by an imaging instrument viewing a point source, and hence is referred to as the *instrument function* or, as is here preferred, the *point spread function* or *psf*. If an image $f(\vec{x})$ is applied to the input of a linear system with psf $h(\vec{x},\vec{x}')$ then the output image $g(\vec{x})$ is (by invoking superposition)

$$g(\vec{x}) = \int_{-\infty}^{\infty} f(\vec{x}')\, h(\vec{x},\vec{x}')\, d\vec{x}' \tag{1.22}$$

An important class of linear systems comprises those for which

$$h(\vec{x}, \vec{x}') = h(\vec{x} - \vec{x}') \tag{1.23}$$

i.e. the output resulting from a delta function at the input is unchanged (but shifted) when the input delta function is shifted. Such systems are called (in image science contexts) *isoplanatic* or, as is here preferred, *point spread invariant* (psi). The more general situation characterized by (1.22) is referred to as *point spread variant* (psv). When (1.23) holds, (1.22) reduces to

$$g(\vec{x}) = \int_{-\infty}^{\infty} f(\vec{x}')\, h(\vec{x} - \vec{x}')\, d\vec{x}' \tag{1.24}$$

which is known as a *convolution integral* (Bates and McDonnell, 1986, §3). The symbol $\odot$ is used to denote convolution, and hence (1.24) is equivalent to

$$g(\vec{x}) = f(\vec{x}) \odot h(\vec{x}) \tag{1.25}$$

Fourier transformation of (1.24) yields the *convolution theorem* (Bates and McDonnell, 1986, §7)

$$f(\vec{x}) \odot h(\vec{x}) \longleftrightarrow F(\vec{u})\, H(\vec{u}) \tag{1.26}$$

The significance of this theorem is, of course, that the process of convolution in image space is simplified to mere multiplication in Fourier space.

The correlation, denoted by the symbol $*$, of two images $f(\vec{x})$ and $h(\vec{x})$ is defined (Goodman, 1968, §7.6) as

$$f(\vec{x}) * h(\vec{x}) = \int_{-\infty}^{\infty} f^*(\vec{x}')\, h(\vec{x} + \vec{x}')\, d\vec{x}' \tag{1.27}$$

Fourier transformation of (1.27) yields the *correlation theorem* (*cf.* Bates and McDonnell, 1986, §7)

$$f(\vec{x}) * h(\vec{x}) \longleftrightarrow F^*(\vec{u})\, H(\vec{u}) \tag{1.28}$$

Correlation of an image with itself is known as *autocorrelation*, with the autocorrelation of $f(\vec{x})$ being denoted by $ff(\vec{x})$. It follows from (1.27) that

$$ff(\vec{x}) = \int_{-\infty}^{\infty} f^*(\vec{x})\, f(\vec{x} + \vec{x}')\, d\vec{x}' \tag{1.29}$$

From (1.28), it is evident that

$$ff(\vec{x}) \longleftrightarrow |F(\vec{u})|^2 \tag{1.30}$$

which is known as the *autocorrelation theorem* (Bates and McDonnell, 1986, §7).

1.5 Sampling

Macroscopic quantities measured in the real world are usually effectively continuous. However, digital computation can only be performed on a finite sized set of numbers.

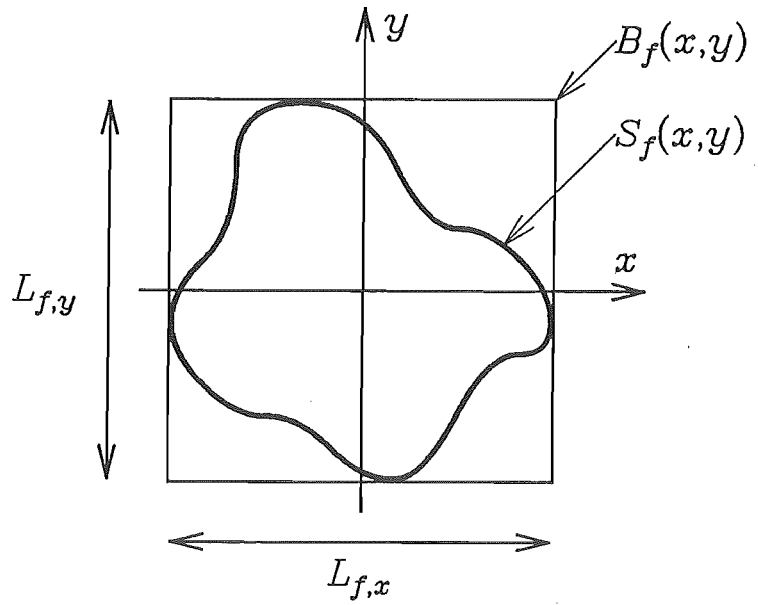


Figure 1.2 The support, image box, and extents of a two-dimensional compact image $f(x, y)$.

The measured quantity must therefore be *sampled* before being processed in a computer. This section describes how one chooses a "sufficient" number of samples to adequately characterize the measured quantity.

All real-world images are of finite size, because any real-world recording device has a finite field of view. It is useful to define the region within which an image $f(\vec{x})$ is nonzero as the *support* of $f(\vec{x})$, denoted by $S_f(\vec{x})$. Another useful concept is the *image box*, which is defined as the region within the smallest rectangle that just encloses the support, and is denoted by $B_f(\vec{x})$. The length of the side of $B_f(\vec{x})$ in the x_k direction is termed the *extent* of $f(\vec{x})$ in that direction, and is denoted by L_{f,x_k}, or simply L_{x_k} if no ambiguity is caused thereby. An image of finite-sized support, and also of finite amplitude, i.e.

$$|f(\vec{x})| < \infty \quad \forall \vec{x}\,, \tag{1.31}$$

is said to be *compact* (Bates, 1982b). Figure 1.2 represents a two-dimensional compact image (which has nonzero value within the thick curve) and its corresponding support, image-box, and extents.

In the following discussion, only two-dimensional images are considered, because these are the major concern of this thesis. The extension to a greater number of dimensions is straightforward. It is convenient to define the *periodic image* $p(x, y)$ to consist of identical versions of a compact image $f(x, y)$ repeated contiguously throughout all of image space, as

$$p(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x + mL_x,\, y + nL_y) \tag{1.32}$$

Note that since $f(x, y)$ is compact, it can be defined in terms of $p(x, y)$ by

$$f(x,y) = \begin{cases} p(x,y) & (x,y) \in B_f(x,y) \\ 0 & (x,y) \notin B_f(x,y) \end{cases} \tag{1.33}$$

Since $p(x,y)$ is (by definition) periodic, with periods L_x and L_y in the x and y directions respectively, it can be represented (in the limit) by a *trigonometric Fourier series* as (*cf.* Brigham, 1974, §5.1)

$$p(x,y) = \frac{1}{L_x L_y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F_{m,n}\, e^{-i2\pi(mx/L_x + ny/L_y)} \tag{1.34}$$

where $F_{m,n}$ are complex coefficients, known as *Fourier coefficients*, whose values are given by

$$\begin{aligned} F_{m,n} &= \int_0^{L_y} \int_0^{L_x} p(x,y)\, e^{i2\pi(mx/L_x + ny/L_y)} dx\, dy & (1.35a) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\, e^{i2\pi(mx/L_x + ny/L_y)} dx\, dy & (1.35b) \end{aligned}$$

where (1.35b) follows from (1.35a) due to (1.33). Comparing (1.35) with (1.14) reveals that

$$F_{m,n} = F(\frac{m}{L_x}, \frac{n}{L_y}) \tag{1.36}$$

Since $f(x,y)$ can be obtained straightforwardly from $p(x,y)$ by (1.33), the significance of (1.36) is that a two-dimensional compact image can be represented exactly by samples of its spectrum, where the sample spacings in the u and v directions are the reciprocals of the extents of the image in the x and y directions respectively. This statement is the essence of the two-dimensional *sampling theorem* (Bates and McDonnell, 1986, §10). The sample spacings given by (1.36) are referred to as the *Nyquist spacings*, and samples spaced by this amount are called *Nyquist samples*. Since the Nyquist samples exactly characterize the image $f(x,y)$, they must also exactly characterize the spectrum $F(u,v)$. To see how this occurs, it is first noted that Fourier transformation of (1.34) yields

$$P(u,v) = \frac{1}{L_x L_y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F_{m,n}\, \delta(u - \frac{m}{L_x})\, \delta(v - \frac{n}{L_y}) \tag{1.37}$$

demonstrating that the spectrum of $p(x,y)$ consists of a delta function (as defined in (1.6)) at each sampling point, the strength of which is proportional to the value of $F(u,v)$ at that point. Now $f(x,y)$ is obtained from $p(x,y)$ by simply isolating a single period of $p(x,y)$, as in (1.33). For the sake of simplicity, it is assumed that $B_f(x,y)$ is centred on the origin of image space, so that (1.33) is equivalent to

$$f(x,y) = p(x,y)\, \mathrm{rect}(\frac{x}{L_x})\, \mathrm{rect}(\frac{y}{L_y}) \tag{1.38}$$

where the rect function is as defined in (1.7). Fourier transformation of (1.38) yields

$$\begin{aligned} F(u,v) &= P(u,v) \odot [\, L_x L_y\, \mathrm{sinc}(L_x u)\, \mathrm{sinc}(L_y v)\,] \\ &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F_{m,n}\, \mathrm{sinc}(L_x\, u - m)\, \mathrm{sinc}(L_y\, v - n) \end{aligned} \tag{1.39}$$

which states that $F(u,v)$ is obtained from its samples as a summation of sinc functions (as defined in (1.7)), each centred at a sample point and scaled by the corresponding sample value.

It is now instructive to consider the effect of choosing sample spacings other than the Nyquist spacings. To do so it is necessary to define the *modified periodic image* $p'(x,y)$ as

$$p'(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F'_{m,n}\, e^{-i2\pi(mx/L'_x + ny/L'_y)} \tag{1.40}$$

where

$$F'_{m,n} = F(\frac{m}{L'_x}, \frac{n}{L'_y}) \tag{1.41}$$

Thus $p'(x,y)$ is the periodic image obtained when the sample spacings in the u and v directions are $1/L'_x$ and $1/L'_y$ respectively. The periods of $p'(x,y)$ are thus L'_x and L'_y in the x and y directions respectively. Consider first the case where the sample spacings are less than the Nyquist spacings i.e.

$$\frac{1}{L'_x} < \frac{1}{L_x} \quad \text{and} \quad \frac{1}{L'_y} < \frac{1}{L_y} \tag{1.42}$$

in which case $F(u,v)$ is said to have been *oversampled.* The periods of $p'(x,y)$ in the x and y directions are now greater than L_x and L_y, and so each individual period of $p'(x,y)$ contains $f(x,y)$ in its centre, and is zero outside. Each period of $p'(x,y)$ is said to be *packed with zeros* in the part of the period outside the part occupied by the support of $f(x,y)$. Note that in this case $f(x,y)$ can be obtained from $p'(x,y)$ by replacing $p(x,y)$ with $p'(x,y)$ in (1.33). If, on the other hand, the sample spacings are less than the Nyquist spacings (in which case $F(u,v)$ is said to have been *undersampled*), then the periods of $p'(x,y)$ in the x and y directions are less than L_x and L_y respectively. Thus each individual period of $p'(x,y)$ consists of overlapped versions of $f(x,y)$. This overlapping is known as *aliasing.* Note that if $p'(x,y)$ is aliased, then it is no longer possible to recover $f(x,y)$ by replacing $p(x,y)$ with $p'(x,y)$ in (1.33).

The preceding discussion is now illustrated with an example (which is one-dimensional for ease of exposition). A compact image $f(x)$, where $f(x) = \text{tri}(x) + \frac{1}{2}\text{tri}(\frac{x}{2})$ is shown in Figure 1.3a. Its spectrum is $F(u) = \text{sinc}^2(u) + \text{sinc}^2(2u)$ (which is purely real), and this is shown in Figure 1.3b. Figure 1.3c shows the periodic image $p(x)$, and the vertical lines in Figure 1.3d represent the Fourier coefficients of $p(x)$ (i.e. the values of F_m in (1.34)). Alternatively, Figure 1.3d can be viewed as the spectrum of $p(x)$, the vertical lines representing delta functions, the magnitude of each of which is indicated by the height of the corresponding line. Figures 1.3e and f show how oversampling the spectrum results in packing with zeros in image space, whilst Figures 1.3g and h show how undersampling the spectrum results in aliasing in image space.

In practice, it is impossible (and indeed unnecessary) to compute an infinite number of harmonics in (1.34). Since any real-world data suffer from contamination and measurement uncertainties, and since $|F_{m,n}|$ tends to decrease with increasing $|m|$ and $|n|$, it is only worthwhile making $|m|$ and $|n|$ large enough to represent $f(x,y)$ to an accuracy equal to the estimated errors in the data (Bates and McDonnell, 1986, §10). Use of a truncated Fourier series gives the *truncated periodic image* $p_t(x,y)$, where

$$p_t(x,y) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} F_{m,n} e^{-i2\pi(mx/L_x + ny/L_y)} \tag{1.43}$$

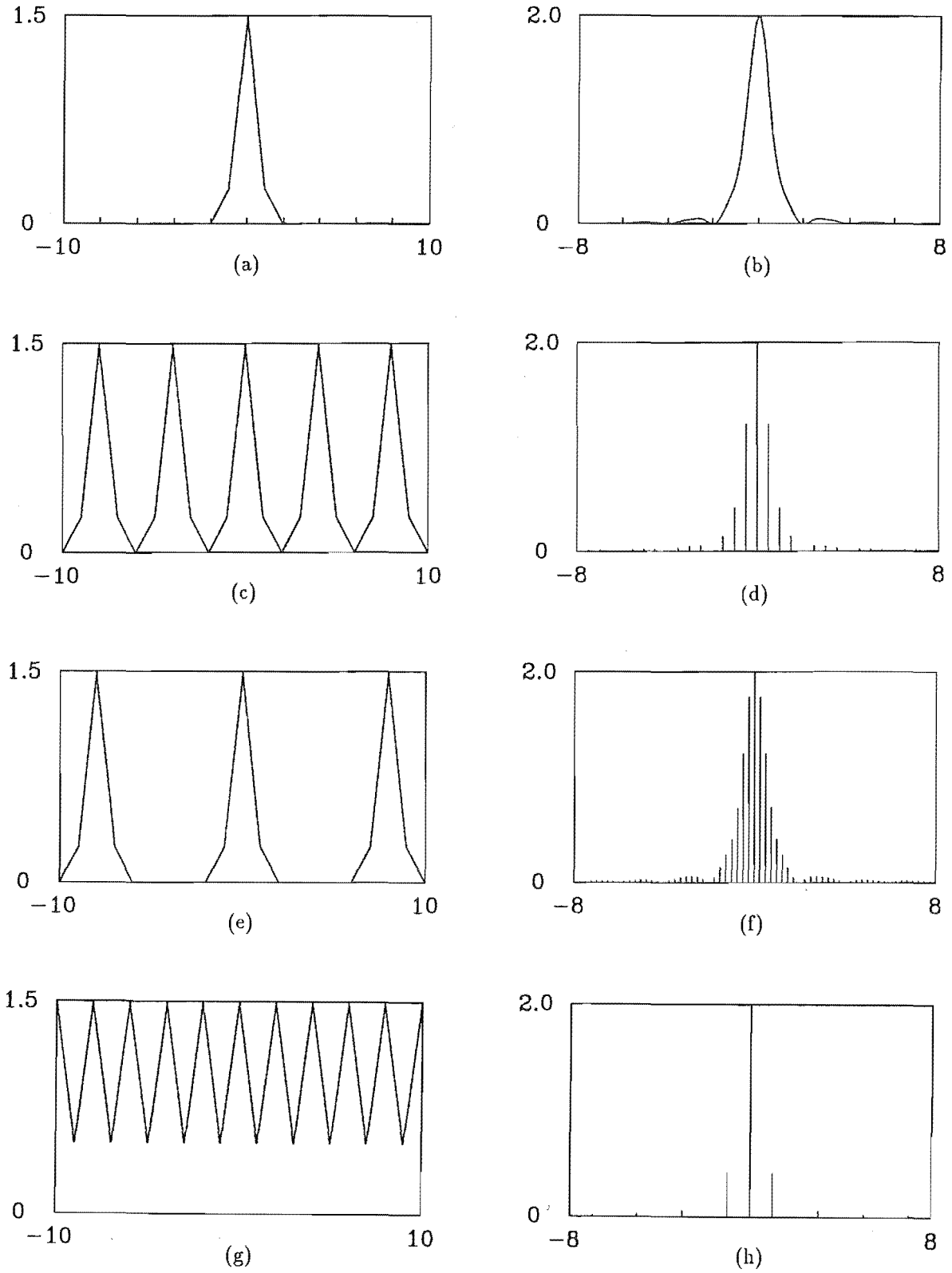


Figure 1.3 Sampling of a one-dimensional compact image. (a) original image $f(x)$, (b) spectrum of (a), (c) periodic image $p(x)$, (d) spectrum of (b) (Nyquist sampling), (e) modified periodic image $p'(x)$ for $L'_x = 2L_x$ (image is packed with zeros), (f) spectrum of (e) (spectrum is oversampled), (g) modified periodic image $p'(x)$ for $L'_x = L_x/2$ (image is aliased), (h) spectrum of (g) (spectrum is undersampled).

Figures 1.4c–h show $p_t(x)$ and $P_t(u)$ for increasing values of M. Note that the agreement between $f(x)$ and $p_t(x)$ within $B\{f(x)\}$ improves as the number of harmonics increases.

It should be noted that, by reversing the roles of image and Fourier space in the preceding discussion, one can show that an image with compact spectrum (of extents L_u and L_v in the u and v directions respectively) can be represented exactly by samples of the image spaced by $1/L_u$ and $1/L_v$ in the x and y directions respectively.

1.6 The Discrete Fourier Transform

In §1.5, it was shown how a compact image can be represented by samples of its spectrum. However, for many of the applications described in this thesis, it is often desirable to perform computations on both an image and its spectrum. This requires that both the image and spectrum be sampled. As explained in §1.5, an image (or spectrum) must be compact if it is to be represented exactly by samples in the other domain. But it is a well known fact (Saxton, 1978, §3.1) that an image and its spectrum cannot both be compact, since the Fourier transform of a compact image is an entire function, and an entire function cannot vanish over any finite region without vanishing everywhere. This dilemma has been addressed by Slepian (1976), who points out that all real-world signals must effectively be both time-limited and band-limited in practice. This follows since there is always some band-limit (or time-limit) which, when applied to a given signal, causes an unmeasurably small change to that signal, and so the signal must be considered effectively band-limited (or time-limited).

In the terminology of image science, the observations made by Slepian (1976) can be rephrased as follows: all real-world images and their spectra must be *effectively compact*, where the *effective support* of an image or spectrum is the region within which it has values that are greater than the estimated noise and uncertainties in the measurements. The point here is that although an image or spectrum may have non-zero value outside its effective support, one has no way of determining whether it does so or not, and so there is no point worrying about it. An image or spectrum which is effectively compact, but not exactly compact, must therefore necessarily suffer from aliasing when represented by samples in the other domain. However, such aliasing is of little consequence, since the level of aliasing is below the level of the noise in the measurements.

When computations are to be performed in both domains, it is useful to be able to relate the samples of an image to the samples of its spectrum. Such a relationship is now developed for the one-dimensional case, where the original Fourier transform pair is

$$f(x) \longleftrightarrow F(u) \tag{1.44}$$

As noted in §1.5, a sampled function can be described as the product of the original function and a series of delta functions space by the sampling interval (known as a *sampling function* (Bracewell, 1978, ch. 5)). Doing so results in a replication of the function in the other domain, which is not surprising since sampling in one domain implies representation by a Fourier series in the other domain. Sampling of the

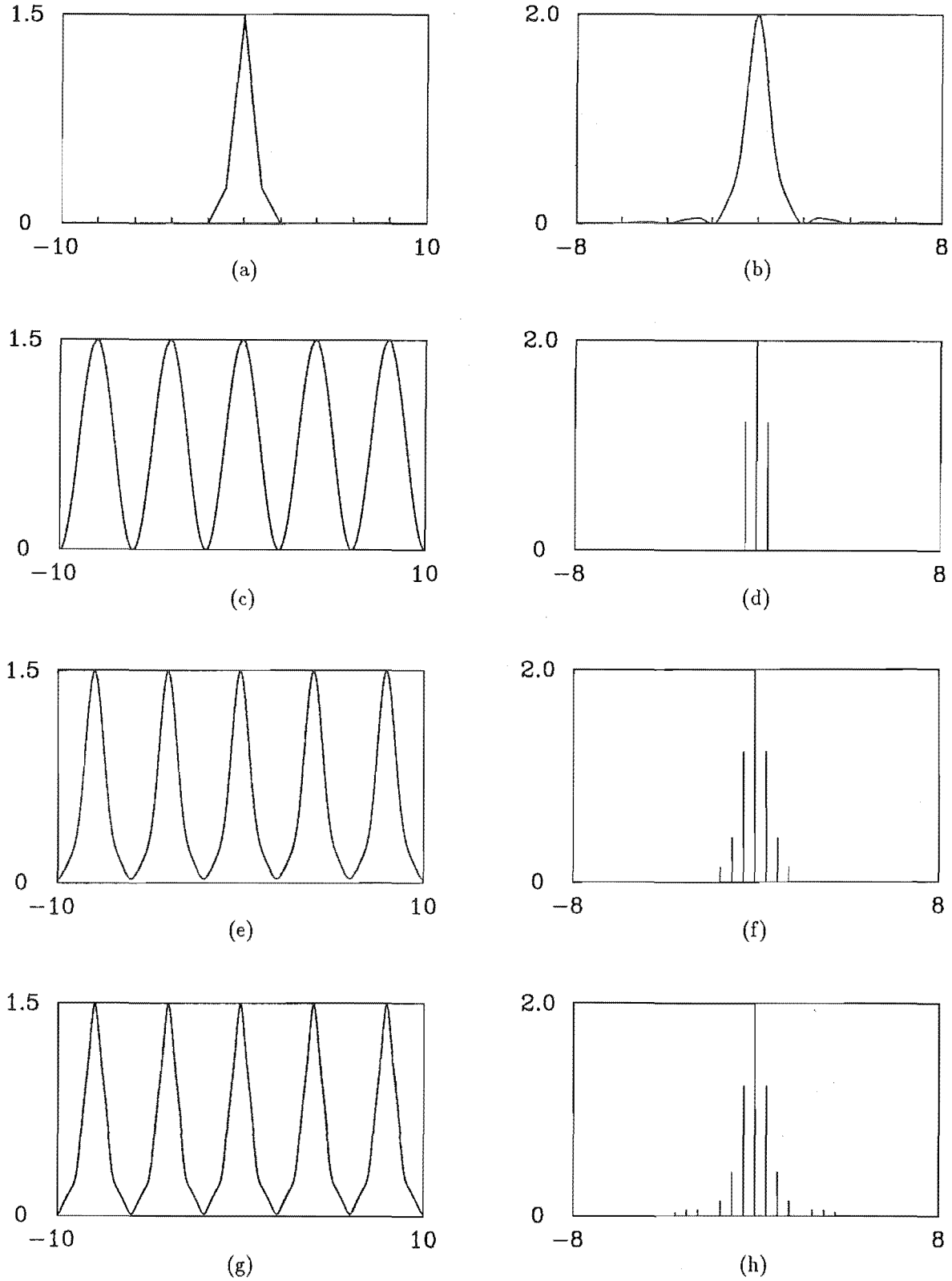


Figure 1.4 Effect of increasing the number of harmonics in a Fourier series representation of a compact image. (a) original image $f(x)$, (b) spectrum of (a), (c) truncated periodic image $p_t(x)$ with $M = 1$, (d) spectrum of (c), (e) truncated periodic image $p_t(x)$ with $M = 3$, (f) spectrum of (e), (g) truncated periodic image $p_t(x)$ with $M = 7$, (h) spectrum of (g).

spectrum (with sampling interval Δ_u) modifies the Fourier transform pair (1.44), which becomes

$$\frac{1}{\Delta_u}\sum_{q=-\infty}^{\infty} f(x-\frac{q}{\Delta_u}) \longleftrightarrow \sum_{m=-\infty}^{\infty} F(m\Delta_u)\,\delta(u-m\Delta_u) \tag{1.45}$$

where (1.26) and the relationship (Bracewell, 1978, ch. 10)

$$\sum_{m=-\infty}^{\infty} \delta(x-\alpha m) \longleftrightarrow \frac{1}{\alpha}\sum_{n=-\infty}^{\infty} \delta(u-\frac{n}{\alpha}) \tag{1.46}$$

have been invoked. Upon sampling image space (with sampling interval Δ_x), the Fourier transform pair becomes

$$\frac{1}{\Delta_u}\sum_{q=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(n\Delta_x)\,\delta(x-n\Delta_x-\frac{q}{\Delta_u}) \longleftrightarrow \frac{1}{\Delta_x}\sum_{p=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} F(m\Delta_u)\,\delta(u-m\Delta_u-\frac{p}{\Delta_x}) \tag{1.47}$$

For this Fourier transform pair to be represented by a finite number of values, it is necessary to ensure that the distance between replicated versions of the sampled image (spectrum) is an integer multiple of the spacing of the image (spectrum) samples, i.e.

$$\frac{1}{\Delta_x\,\Delta_u} = N\,, \quad N \in I^+ \tag{1.48}$$

Once (1.48) is satisfied, the Fourier transform pair (1.47) is amenable to digital computation, since each member of the pair is represented by N discrete values. This can be seen by expressing (1.47) as

$$\frac{1}{\Delta_u}\sum_{q=-\infty}^{\infty}\sum_{n=0}^{N-1} f_n\,\delta(x-n\Delta_x-\frac{q}{\Delta_u}) \longleftrightarrow \frac{1}{\Delta_x}\sum_{p=-\infty}^{\infty}\sum_{m=0}^{N-1} F_m\,\delta(u-m\Delta_u-\frac{p}{\Delta_x}) \tag{1.49}$$

where

$$f_n = \sum_{\alpha=-\infty}^{\infty} f(n\Delta_x-\frac{\alpha}{\Delta_u}) \quad 0 \le n \le N-1 \tag{1.50}$$

and

$$F_m = \sum_{\beta=-\infty}^{\infty} F(m\Delta_u-\frac{\beta}{\Delta_x}) \quad 0 \le m \le N-1 \tag{1.51}$$

It is now desired to relate the N values of F_m to the N values of f_n. This is achieved by substituting LHS (1.49) into (1.14), and performing some further manipulations (see Brigham, 1974, §6.2) to yield

$$F_m = \Delta_x \sum_{n=0}^{N-1} f_n\, e^{i2\pi mn/N} \tag{1.52}$$

which is known as the *discrete Fourier transform* or DFT. By similar reasoning, the *inverse discrete Fourier transform* or IDFT can be derived as

$$f_n = \Delta_u \sum_{m=0}^{N-1} F_m\, e^{-i2\pi mn/N} \tag{1.53}$$

It has thus been shown how a periodic sampled image is related to its periodic sampled spectrum. Just how similar these values are to the samples of the original continuous image and continuous spectrum depends solely on the degree of aliasing in each domain, resulting from sampling in the other domain. This point must always be borne in mind when applying the DFT to real-world data. Note that the exact relationships between the values of the sampled periodic image (and spectrum) and the continuous image (and spectrum) are described by (1.50) and (1.51).

It is instructive to illustrate the points made in this section with an example. Figures 1.4a and b show the original compact image and its spectrum. Note that although the spectrum is not exactly compact, it does indeed fall off rapidly with increasing u. Initially, the spectrum is sampled, resulting in replication of $f(x)$ in image space, but note that aliasing does not occur since $f(x)$ is compact and $F(u)$ is sampled at the Nyquist rate (i.e $\Delta_u = \frac{1}{2}$). Next, the replicated image is sampled (with sampling interval $\Delta_x = \frac{1}{4}$), resulting in replication of the sampled spectrum. Note that this causes aliasing in Fourier space, since $F(u)$ is not compact. The effect of aliasing can be reduced by choosing a smaller sampling interval in image space, as shown in Figures 1.5g and h, where $\Delta_x = \frac{1}{8}$. It is evident in this case that the agreement between the values of the sampled periodic spectrum and the original continuous spectrum improves as Δ_x is reduced. Note that each period of the periodic sampled image consists of $N = 1/\Delta_x\Delta_u$ values, and it is these N values which are related by the DFT.

It is customary (Brigham, 1974, §6.3) to omit the scale factor in the forward DFT, and lump it in with that for the inverse DFT, as indicated in

$$G[m] = \sum_{n=0}^{N-1} g[n]\, e^{i2\pi mn/N} \tag{1.54}$$

and

$$g[n] = \frac{1}{N} \sum_{m=0}^{N-1} G[m]\, e^{-i2\pi mn/N} \tag{1.55}$$

where the notation introduced in §1.2 for discrete variables has been invoked. The two-dimensional forward and inverse DFT are given by (*cf.* Bracewell, 1978, ch. 18),

$$G[m_1, m_2] = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} g[n_1, n_2]\, e^{i2\pi(m_1 n_1/N_1 + m_2 n_2/N_2)} \tag{1.56}$$

and

$$g[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} G[m_1, m_2]\, e^{-i2\pi(m_1 n_1/N_1 + m_2 n_2/N_2)} \tag{1.57}$$

Direct evaluation of (1.54) for each m requires N^2 complex multiplications and $N(N-1)$ complex additions. However, a computational saving can be achieved by invoking the *fast Fourier transform algorithm* (FFT) (Cooley and Tukey, 1965; Brigham, 1974, ch. 10), which essentially decomposes the DFT into a number of smaller DFTs. Most FFT programs require N to be a positive power of 2, and typically require $(N \log_2 N)/2$ complex multiplications and $N \log_2 N$ complex additions (Brigham, 1974, ch. 10). It is evident that the computational saving of the FFT

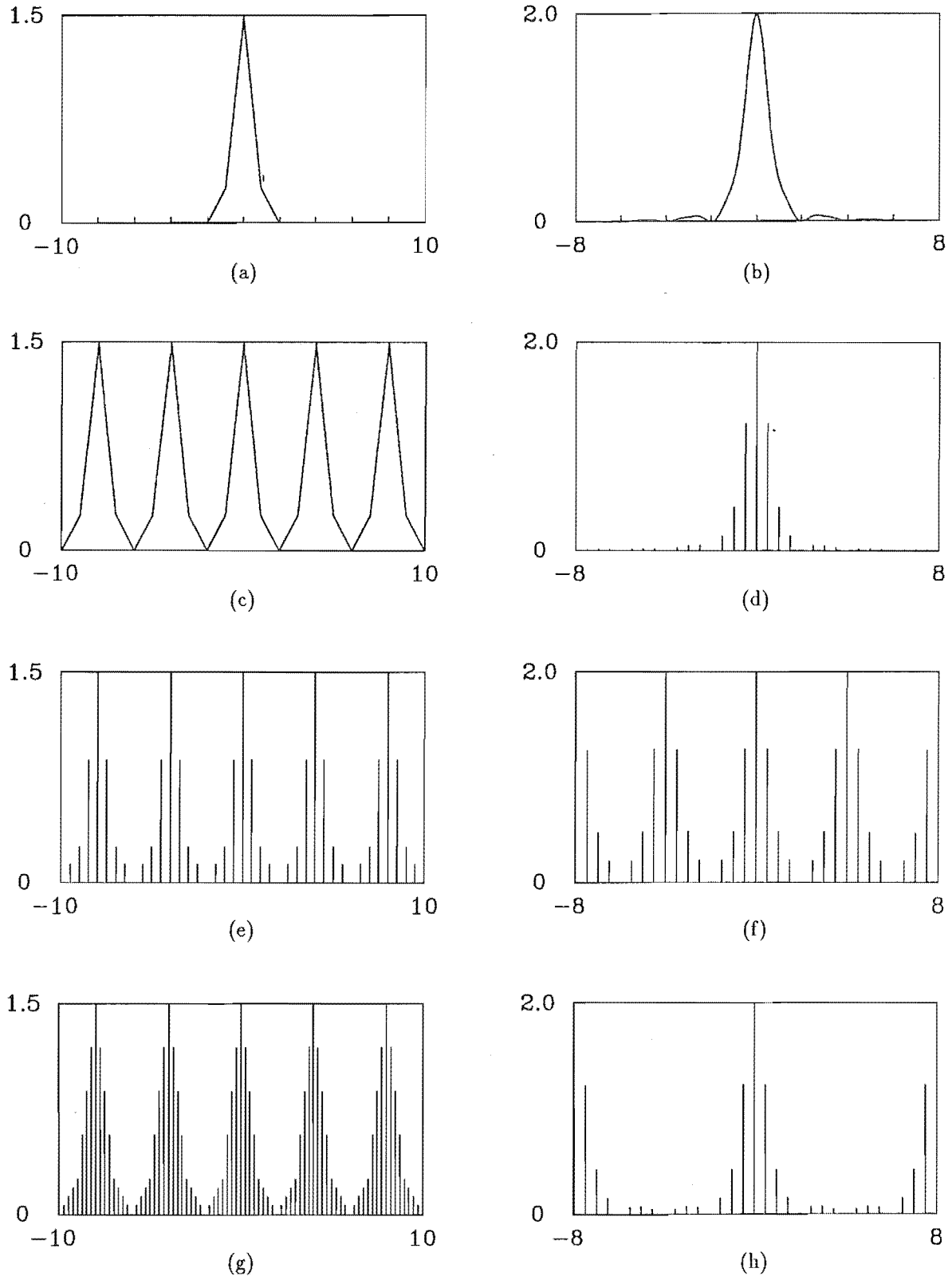


Figure 1.5 Relationship between the continuous Fourier transform and the DFT. (a) original image, (b) spectrum of (a), (c) replicated image, resulting from (d) sampled spectrum ($\Delta_u = \frac{1}{2}$), (e) sampled replicated image ($\Delta_x = \frac{1}{4}$), (f) replicated sampled spectrum, resulting from (e), (g) as for (e) but ($\Delta_x = \frac{1}{8}$), (h) replicated sampled spectrum, resulting from (g).

over the DFT is significant if N is large, and is especially so for two-dimensional data. Furthermore, the reduction in the number of operations significantly reduces the roundoff error when N is large.

1.7 The Z-Transform and Zeros

It is shown in §1.6 how an image of finite extent can be represented to within any desired degree of accuracy by a finite-sized set of samples of the image. Alternatively, the same image can be represented by a finite-sized set of samples of its spectrum. Numerous other methods for representing such images exist, and this section describes one method which is made use of in subsequent chapters.

On sampling a continuous one-dimensional image $f(x)$ with sampling interval Δ_x, the resulting sampled image can be represented by

$$f_s(x) = \sum_{m=0}^{M-1} f[m]\,\delta(x - m\Delta_x) \tag{1.58}$$

where the subscript "s" indicates that the image is a sampled version of $f(x)$. The spectrum of the sampled image is then

$$F_s(u) = \sum_{m=0}^{M-1} f[m]\,e^{i2\pi m\Delta_x u} \tag{1.59}$$

It is useful to generalize the functional dependence of F_s from the real variable u to the complex variable w, where

$$w = u + iv \tag{1.60}$$

Thus (1.59) becomes

$$F_s(w) = \sum_{m=0}^{M-1} f[m]\,e^{i2\pi m\Delta_x w} \tag{1.61}$$

$F_s(u)$ is thereby *analytically continued* (Churchill *et al.*, 1974, ch. 12) from the real u axis into the whole w plane. On defining ζ to be

$$\zeta = e^{i2\pi m\Delta_x w} \tag{1.62}$$

(1.61) becomes

$$F_s(w) = \mathcal{F}(\zeta) = \sum_{m=0}^{M-1} f[m]\,\zeta^m \tag{1.63}$$

which is a polynomial in ζ of degree $M-1$, and is known as the *z-transform* (*cf.* Papoulis, 1980, §3.1) or *z-spectrum* of $f[m]$. Note the convention adopted whereby variables in z-space (the space in which z-transforms exist) are denoted by the same letter as that of the sampled image, but in upper-case calligraphic type. If (1.63) is equated to zero, the solutions of

$$\mathcal{F}(\zeta) = 0 \tag{1.64}$$

are known as the *roots* or *zeros* of $\mathcal{F}(\zeta)$. By the fundamental theorem of algebra (Churchill *et al.*, 1974, §55), it can be shown that $\mathcal{F}(\zeta)$ possesses exactly $M-1$ zeros

(although some zeros may be repeated). Furthermore, $\mathcal{F}(\zeta)$ can be represented by (Requicha, 1980)

$$\mathcal{F}(\zeta) = f[M-1] \prod_{j=1}^{M-1} (\zeta - \zeta_j) \tag{1.65}$$

The significance of (1.65) is that $\mathcal{F}(\zeta)$ (and hence $f[m]$) can be represented exactly (to within a complex scale factor) by its zeros. The original sampled image can be recovered (to within a complex scale factor) from the zeros of $\mathcal{F}(\zeta)$ by evaluating $\mathcal{F}(\zeta)$ by (1.63) for M values of ζ equally spaced around the unit circle in the ζ-plane. Denoting such values by $\zeta^{(p)}$, where

$$\zeta^{(p)} = e^{i2\pi p/M} \qquad p = 0, 1, \ldots, M-1 \tag{1.66}$$

and the corresponding values of $\mathcal{F}(\zeta)$ by $\mathcal{F}^{(p)}$, it is evident from (1.63) that

$$\mathcal{F}^{(p)} = \sum_{m=0}^{M-1} f[m]\, e^{i2\pi mp/M} \tag{1.67}$$

Comparison with (1.54) reveals that $\mathcal{F}^{(p)}$ is the DFT of $f[m]$, and hence $f[m]$ may be obtained from $\mathcal{F}^{(p)}$ by simply performing an inverse DFT.

The zeros of $\mathcal{F}(\zeta)$ are denoted by $\mathcal{Z}\{\mathcal{F}(\zeta)\}$, i.e.

$$\mathcal{Z}\{\mathcal{F}(\zeta)\} = \{\zeta_1, \zeta_2, \ldots, \zeta_{M-1}\} \tag{1.68}$$

It should be mentioned that the z-transform of a two-dimensional image is given by

$$\mathcal{F}(\zeta, \gamma) = \sum_{m=0}^{M} \sum_{n=0}^{N} f[m,n]\, \zeta^m\, \gamma^n \tag{1.69}$$

The properties of zeros of two-dimensional images are discussed in detail §2.4.2.

Chapter 2

Phase Problems and Blind Deconvolution Problems

This chapter introduces the two image reconstruction problems which are the concern of this thesis; namely the phase problem (§§2.1–2.5) and the blind deconvolution problem (§§2.6–2.8). It may seem more sensible to discuss the blind deconvolution problem before the phase problem, since (as is shown in §2.6) the latter is merely a special case of the former. This is not done here, however, because the major concern of this thesis is with the phase problem.

The problem of recovering an image (or, equivalently, its spectral phase) from knowledge of its spectral magnitude alone, which is commonly referred to as the phase problem, is introduced in §2.1. The somewhat disheartening (from a phase retrieval viewpoint) observation that the majority of the information content of an image appears to be contained in the spectral phase, a characteristic of images known as phase dominance, is demonstrated in §2.2. The manner in which the spectral phase is constrained by the spectral magnitude, when the image is known to satisfy certain simple constraints, is described in §2.3. The essential question of uniqueness of solutions to phase problems is discussed in §2.4, for both one-dimensional and multi-dimensional images, and also for situations in which the given magnitude data are contaminated. The introduction to the phase problem concludes, in §2.5, with a description of various commonly encountered phase problems.

The problem of recovering two images from their convolution, which has become known as the blind deconvolution problem, is introduced in §2.6. The question of uniqueness of solutions to the blind deconvolution problem is considered in §2.7, and finally, two examples of practical blind deconvolution problems are given in §2.8.

2.1 The Phase Problem

The desire to reconstruct an image, when the magnitude (or, equivalently, the intensity) of its spectrum is given, but the spectral phase is unknown (or only known inaccurately), presents what has become known as a *phase problem*. A situation in which only the spectral magnitude is given constitutes a *pure phase problem*, whereas partial knowledge of the spectral phase in addition to knowledge of the spectral magnitude constitutes a *partial phase problem* (Bates and Mnyama, 1986). Consideration

is only given to pure phase problems in this thesis, but it should be emphasized that the methods described for solving pure phase problems can be usefully applied to partial phase problems.

Phase problems occur in many remote sensing/probing situations, where an object's visibility is obtained from measurements of wave motion of some kind. Measurement of the visibility magnitude is often more straightforward than measurement of the visibility phase, for the reasons which follow. Devices to measure the intensity only of wave motions of various kinds tend to be relatively simple to construct. Examples of common intensity detectors are the envelope detector for radio-frequency applications, and photo-detectors and photographic film for use at optical frequencies. Measurement of the phase of some wave motion, on the other hand, requires a reference signal against which the incoming signal must be compared. The technical difficulties encountered in providing such suitable reference signals become increasingly severe as the operating frequency increases.

A further factor complicating phase measurement is that the phase of the sensed wave motion is often distorted (more so than the magnitude) during propagation, or distorted by the imaging instrument. A selection of common examples of practical phase problems is given in §2.5.

It should be noted that if an image $f(\vec{x})$ is shifted in image space, or conjugated and then rotated by 180°, or multiplied by a constant phase factor, then $f(\vec{x})$ can still be recognized in the resulting image. It is thus useful to say that any of the aforesaid variations of $f(\vec{x})$, or combinations of these variations, have the same *image-form* as $f(\vec{x})$ (Bates, 1982a). Thus a particular image-form of $f(\vec{x})$, denoted by $f_{\text{if}}(\vec{x})$, is

$$f_{\text{if}}(\vec{x}) = e^{\alpha} f(s\vec{x} - \vec{\xi}) \tag{2.1}$$

where $s = \pm 1$, $\vec{\xi}$ is an arbitrary constant position vector, and α is a real constant. Fourier transformation of (2.1) yields

$$F_{\text{if}}(\vec{u}) = sF(s\vec{u})e^{i2\pi s\vec{\xi}\cdot\vec{u}+\alpha} \tag{2.2}$$

which shows that any image-form of $f(\vec{x})$ has the same Fourier magnitude as $f(\vec{x})$, which is, of course, the reasoning behind the definition of the image-form. Consequently, when attempting to reconstruct an image from $|F(\vec{u})|$ alone, the best that we can hope to do is to reconstruct an image-form of $f(\vec{x})$.

It is now convenient to define the *general phase problem* as :

> Given samples of $|F(\vec{u})|$, spaced by Δu_k in the u_k direction, for $k = 1, 2, \ldots, K$, reconstruct an image-form of $f(\vec{x})$, subject to any given constraints on $f(\vec{x})$.

All of the specific phase problems discussed in §2.5 are special cases of the general phase problem defined above, differing only in their particular constraints on $f(\vec{x})$, and the spacing of the samples of $|F(\vec{u})|$. Most of the examples presented in subsequent chapters of this thesis are concerned with phase problems which are characterized by sample spacings satisfying

$$\Delta u_k \leq \frac{1}{2L_{f,x_k}} \tag{2.3}$$

where L_{f,x_k} is defined in §1.5. Phase problems satisfying (2.3) are called *Fourier phase problems* by Bates and McDonnell (1986, §20). The significance of (2.3) is that it allows the autocorrelation $ff(\vec{x})$ of $f(\vec{x})$ to be computed immediately from the given samples of $|F(\vec{u})|$. This follows from the extents of $ff(\vec{x})$ being related to those of $f(\vec{x})$ by (Bates and McDonnell, 1986, §7)

$$L_{ff,x_k} = 2L_{f,x_k} \tag{2.4}$$

Since $ff(\vec{x}) \longleftrightarrow |F(\vec{u})|^2$ (§1.4), it is evident that (2.3) is equivalent to requiring that the spectral intensity $|F(\vec{u})|^2$ is sampled in accordance with the sampling theorem, in which case $ff(\vec{x})$ can be reconstructed from samples of $|F(\vec{u})|^2$ satisfying (2.3).

It should be noted that the crystallographic phase problem (discussed further in §2.5.1), in which the image consists of periodic repetitions of the unit cell, has a spectrum consisting of discrete points, spaced by $1/L_{f,x_k}$, for the reasons outlined in §1.5. Data for the crystallographic phase problem thus cannot satisfy (2.3), and so most of the procedures discussed in this thesis are inapplicable to its solution.

2.2 Phase Dominance

It is instructive, at this stage of the discussion of phase problems, to mention the different manner in which the information in a signal is coded by the spectral magnitude and spectral phase. A number of authors, in fields such as x-ray crystallography (Ramachandran and Srinivasan, 1970), and image and speech processing (Oppenheim and Lim, 1981; Bates and Mnyama, 1986, §III.E), have reported that the spectral phase appears to be more important than the spectral magnitude in determining the general structure of a signal (or image). That is to say, the major features of a signal (or image) tend to be preserved if only the phase is retained, but the same cannot be said of the magnitude.

The reason for these remarks is that the spectral phase describes the interrelation between spatial frequency components of an image, which in turn determine the relative positioning of features of the image. The spectral magnitude, on the other hand, merely describes the strengths of the spatial frequency components, which would be expected to have a lesser effect on the features of the image, and how they are perceived by a human observer.

An illustration of phase dominance is presented in Figure 2.1. Each pixel of the images presented in Figure 2.1 is displayed as one of 36 shades of grey. The greyscale is chosen such that the minimum and maximum valued pixel of each image are displayed as black and white respectively. This convention is adhered to throughout this thesis, unless it is stated otherwise. Note particularly that each of the composite images shown in Figure 2.1c,d appears most similar to that original image from which its spectral phase is derived.

Upon considering the dominance of spectral phase over spectral magnitude in determining the form of an image, the reconstruction of an image from its spectral magnitude alone may at first sight appear to be a hopeless task. However, it transpires that, in many practical situations, the spectral phase can be severely constrained by the spectral magnitude. That is to say, the number of possible choices of phase

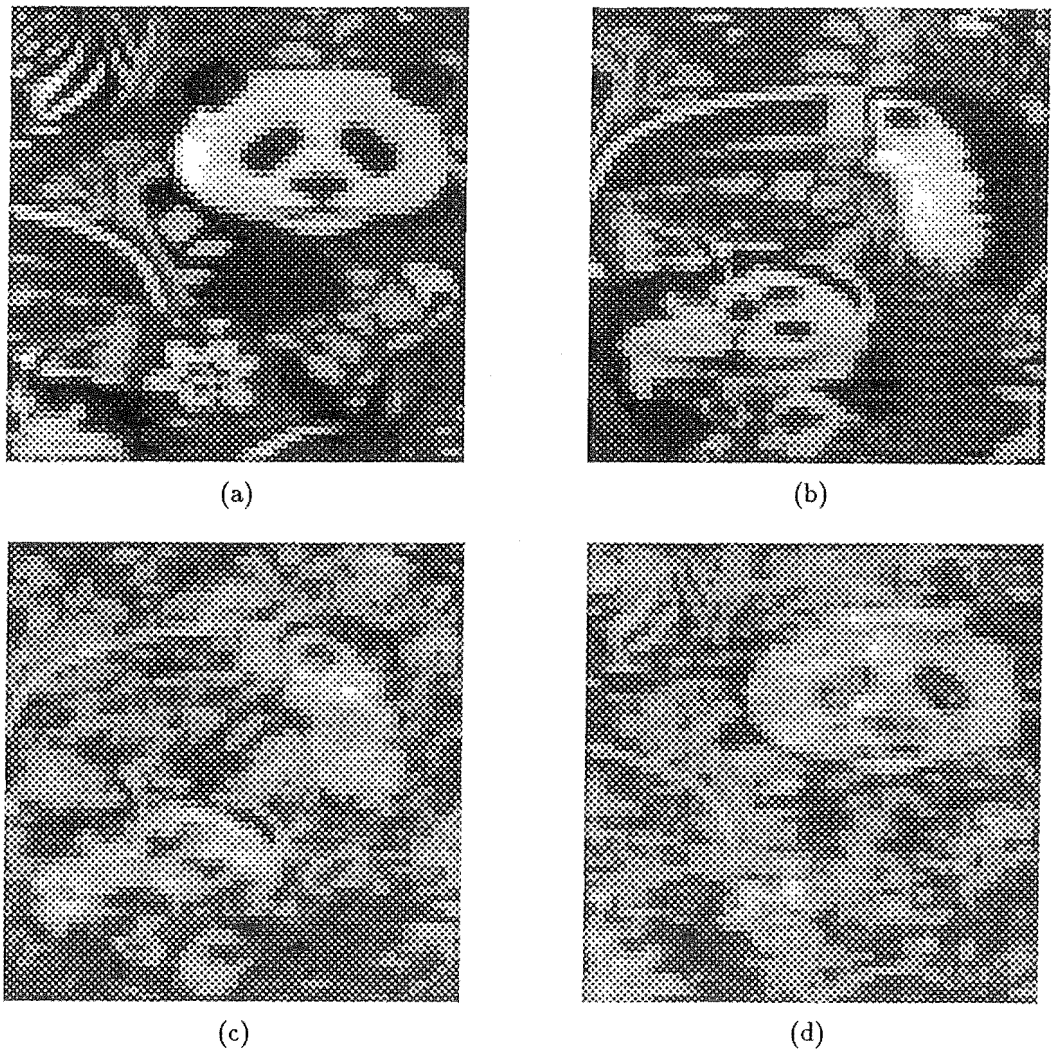


Figure 2.1 Illustration of phase dominance. (a) first original image $f_1(\vec{x})$, (b) second original image $f_2(\vec{x})$, (c) composite image $c_1(x,y)$, computed by $c_1(x,y) = \mathrm{FT}^{-1}\{|F_1(u,v)|\exp(i\mathrm{Ph}\{F_2(u,v)\})\}$, and (d) composite image $c_2(x,y)$, as computed by $c_2(x,y) = \mathrm{FT}^{-1}\{|F_2(u,v)|\exp(i\mathrm{Ph}\{F_1(u,v)\})\}$.

distributions, given a particular magnitude distribution, can be usefully small in many instances. If all the possible choices of phase distributions result in the same image-form, then the magnitude is said to uniquely specify the image-form. The particular situations in which the image-form is uniquely specified by the spectral magnitude are discussed in the next two sections.

2.3 Relationship Between Magnitude and Phase

Since the spectral phase is most important in determining the general form of an image, as demonstrated in the previous section, it is informative to discuss how the spectral phase is constrained by the spectral magnitude, given some *a priori* knowledge concerning the image. If no *a priori* knowledge about the image is available, then any phase function $\hat{\Phi}(\vec{u})$ can be attached to the given spectral magnitude $|F(\vec{u})|$,

and cannot be preferred to any other choice of $\hat{\Phi}(\vec{u})$. However, if one does have some *a priori* knowledge concerning the image $f(\vec{x})$ then it is likely that the resultant image $\hat{f}(\vec{x})$, where

$$\hat{f}(\vec{x}) = \mathrm{FT}^{-1}\{|F(\vec{u})|e^{i\hat{\Phi}(\vec{u})}\} \tag{2.5}$$

will violate the given constraints if the choice of $\hat{\Phi}(\vec{u})$ is arbitrary. For example, if $f(x)$ is both positive and compact, the image $\hat{f}(x)$ obtained from (2.5) with $\hat{\Phi}(u)$ chosen pseudo-randomly is neither positive nor compact, in general.

Various commonly encountered image-domain constraints, and how they cause the spectral phase to be constrained by the spectral magnitude, are now discussed. If a one-dimensional image $f(x)$ satisfies

$$f(x) = 0 \quad \text{for } x \leq 0 \tag{2.6}$$

then the real and imaginary parts of its spectrum satisfy the following Hilbert transform relations (also known as dispersion relations) (Saxton, 1978, §3.5)

$$\mathrm{Re}\{F(u)\} = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\mathrm{Im}\{F(u')\}}{u'-u}du' \tag{2.7}$$

$$\mathrm{Im}\{F(u)\} = -\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\mathrm{Re}\{F(u')\}}{u'-u}du' \tag{2.8}$$

where the Cauchy principal value of the integrals is taken to allow for the divergence at $u = u'$. Evidently, for a given $|F(u)|$, the possible choices of $\hat{\Phi}(u)$ that satisfy (2.7) and (2.8) are limited.

A constraint on the image that is more restrictive than (2.6) is that the image has finite extent, i.e.

$$f(x) = 0 \quad \text{for } |x| > \frac{L_x}{2} \tag{2.9}$$

As shown in §1.5, the spectrum in such cases must satisfy

$$F(u) = \sum_{n=-\infty}^{\infty} F(n/L_x)\,\mathrm{sinc}\,(L_x u - n) \tag{2.10}$$

Once again, it can be seen that the choices of possible $\hat{\Phi}(u)$ that satisfy (2.10) for a given $|F(u)|$, are severely limited.

A further constraint on the image, often encountered for the reasons outlined in §1.3, is that the image must be positive. The manner in which the spectral magnitude constrains the spectral phase, given a positivity constraint on the image, is possibly best explained by the approach taken by Cadzow and Sun (1986). If an image $f(x)$ is positive, then

$$\int_{-\infty}^{\infty} f(x)\,|w(x)|^2\,dx \geq 0 \quad \text{if } f(x) \geq 0 \tag{2.11}$$

for any (possibly complex) choice of $w(x)$. Substitution of $f(x)$ and $w(x)w^*(x)$, expressed in terms of their respective Fourier transforms, into (2.11) yields

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u_1-u_2)\,W(u_1)\,W^*(u_2)\,du_1\,du_2 \geq 0 \quad \text{if } f(x) \geq 0 \tag{2.12}$$

Note that the above inequality holds for any choice of $W(u)$. To make clear how (2.12) constrains $\hat{\Phi}(u)$ when $|F(u,v)|$ is known, (2.12) is written in discrete form as

$$\sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} F[p_1 - p_2]\, W[p_1]\, W^*[p_2] \geq 0 \quad \text{if } f[m] \geq 0 \tag{2.13}$$

As a simple example, the choice of $W[0] = e^{i\alpha}$, $W[1] = e^{i\beta}$ (where α and β are both real constants), and the remaining terms of $W[p]$ all zero, results in the following constraints on $F[p]$, for various choices of α and β :

$$-F[0] \leq \mathrm{Re}(F[1]) \leq F[0] \quad \text{and} \quad -F[0] \leq \mathrm{Im}(F[1]) \leq F[0] \tag{2.14}$$

Whilst (2.14) does not constrain $\hat{\Phi}[1]$, since $|F[0]|$ always exceeds $|F[1]|$ if $f[m]$ is positive, further constraints on $F[p]$ can, of course, be derived for other choices of $W[p]$. It appears that the inequalities become more powerful as the extent of $W[p]$ is made larger.

It can be shown (Cadzow and Sun, 1986) that, for all choices of $W[p]$, (2.13) is equivalent to the requirement that

$$\begin{vmatrix} F[0] & F^*[1] & \cdots & F^*[n] \\ F[1] & F[0] & \cdots & F^*[n-1] \\ \vdots & \vdots & \ddots & \vdots \\ F[n] & F[n-1] & \cdots & F[0] \end{vmatrix} \geq 0 \tag{2.15}$$

for all values of n satisfying $1 \leq n \leq N$, when $F[m]$ comprises $2N+1$ samples. The determinantal inequalities of Karle and Hauptman (1950), which have played an important role in the solution of the crystallographic phase problem, are of similar form to (2.15).

Note that the constraint of positivity in image space is an inequality constraint, and hence the equivalent Fourier domain constraint is also an inequality constraint. The extent constraint in image space, on the other hand, being an equality constraint, results in an equality constraint in Fourier space.

The above discussion indicates how certain constraints on the image result in the choices of possible spectral phase being greatly reduced if the spectral magnitude is known. The discussion has been limited to one-dimensional images only. However, the previously mentioned image constraints tend to place even more severe restrictions on the possible choices of phase distributions, for a given magnitude distribution, in the multi-dimensional case. The question of whether the number of choices is then reduced sufficiently to enable a unique phase to be selected is the subject of the next section.

2.4 Uniqueness of Phase Problems

It is of central interest, when one attempts to solve a particular phase problem, to know whether or not the problem possesses a unique solution. Note that "possesses a unique solution" should be taken to mean (for the reasons given in §2.1) "possesses solutions which all have the same image-form". It transpires that this uniqueness

question for the one-dimensional phase problem, which is discussed in §2.4.1, is fundamentally different from the same question for the multidimensional phase problem (discussed in §2.4.2). The effect of noise on uniqueness of phase problems is addressed in §2.4.3.

2.4.1 One-dimensional Case

The question of uniqueness of the one-dimensional phase problem has been examined by many authors (e.g Walther, 1963; Hofstetter, 1964; Bates, 1969; Burge *et al.*, 1976), mainly with regard to images that are functions of a continuous variable. The approach taken here, which is based on that of Bruck and Sodin (1979), concerns images that are functions of a discrete variable. Consider a discrete (possibly complex) image $f[m]$, of extent M samples, whose z-transform is $\mathcal{F}(\zeta)$. Since the autocorrelation $a[m]$ of $f[m]$ is

$$a[m] = f[m] \odot f^*[-m] \tag{2.16}$$

the z-transform $\mathcal{A}(\zeta)$ of $a[m]$ is given by (Oppenheim and Schafer, 1975)

$$\mathcal{A}(\zeta) = \mathcal{F}(\zeta)\,\mathcal{F}^*(1/\zeta^*) \tag{2.17}$$

It is evident from (2.17) that $\mathcal{A}(\zeta)$ is zero wherever $\mathcal{F}(\zeta)$ or $\mathcal{F}^*(1/\zeta^*)$ is zero. Recalling the discussion in §1.7, the set of zeros of $\mathcal{A}(\zeta)$ is thus seen to be the union of the set of zeros of $\mathcal{F}(\zeta)$ and the set of zeros of $\mathcal{F}^*(1/\zeta^*)$. Writing the set of zeros of $\mathcal{F}(\zeta)$ (of which there are $M-1$ members) as $\{\zeta_m : m = 1,\ldots,M-1\}$, the set of zeros of $\mathcal{F}^*(1/\zeta^*)$ is simply $\{1/\zeta_m^* : m = 1,\ldots,M-1\}$. Since $1/\zeta_m^* = \zeta_m/|\zeta_m|^2$, it follows that the zeros of $\mathcal{A}(\zeta)$ occur in pairs, each member of which lies on opposite sides of (or possibly both lie upon) the unit circle in the ζ-plane. An image whose autocorrelation is $a[m]$ must have a z-transform which possesses one zero from each of the $M-1$ aforesaid pairs. But without further *a priori* information, one cannot know which zero of each pair corresponds to that of the original image.

It is convenient to denote a solution to the phase problem by $\hat{f}[m]$, with z-transform $\hat{\mathcal{F}}(\zeta)$, the set of zeros of which is $\{\zeta'_m : m = 1,\ldots,M-1\}$, where ζ'_m is either ζ_m or $1/\zeta_m^*$. It is evident that there are 2^{M-1} possible combinations of the $\{\zeta'_m : m = 1,\ldots,M-1\}$, and hence 2^{M-1} possible solutions $\hat{f}[m]$. It is said that the problem has an *ambiguity* of 2^{M-1}. This process of generating all possible combinations of zeros is often called *zero-flipping* (Napier, 1971, §5.1.3). Since half of the possible $\hat{f}[m]$ are mirror images of others (because flipping all the zeros of a particular $\hat{f}[m]$ produces $\hat{f}[-m]$) which have (by definition) the same image-form, the ambiguity is reduced to 2^{M-2}. Furthermore, if $f[m]$ is real, the zeros of $\mathcal{F}(\zeta)$ are either real or occur in conjugate pairs (i.e. if ζ_m is a zero if $\mathcal{F}(\zeta)$, and is complex, then so is ζ_m^*). The zeros of $\mathcal{A}(\zeta)$ thus also occur in conjugate pairs, and must be flipped in pairs to ensure that the resulting image is real. In such cases, the ambiguity is reduced to $2^{(M-1)/2-1}$ (assuming no zeros are real). Further reduction of the ambiguity results if any zeros of $\mathcal{F}(\zeta)$ are real, since flipping a real zero does not change $\hat{f}[m]$.

If, in a particular situation, it is known that the image must satisfy certain additional constraints, then the ambiguity in the solution of the phase problem may be

further reduced, or possibly eliminated. For example, if the image is known *a priori* to be positive, then any solutions obtained by zero-flipping that are not everywhere positive can be discarded (Bates, 1969). If the image is known to be both real and even (i.e. $f(x) = f^*(x) = f(-x)$) then it can be uniquely recovered from its autocorrelation (Napier, 1971, §5.1.3; Lawton, 1981), because the zeros of $\mathcal{F}(\zeta)$ are identical to those of $\mathcal{F}^*(1/\zeta^*)$. All zeros of $\mathcal{A}(\zeta)$ are hence double zeros, and thus $\mathcal{F}(\zeta)$ can be recovered unambiguously from $\mathcal{A}(\zeta)$ by retaining one zero from each of the double zeros of $\mathcal{A}(\zeta)$. An image consisting of two or more separated parts (each part possessing finite extent) is usually uniquely recoverable from its Fourier transform magnitude (Napier, 1971, §7.5; Greenaway, 1977; Crimmins and Fienup, 1981; Crimmins and Fienup, 1983).

It has been noticed (Fienup, 1978) that although the number of possible solutions to the one-dimensional phase problem is large, the solutions tend to fall into a smaller number of groups, the members of each group appearing similar. This is because the general form of an image is determined primarily by its low spatial-frequency components, which in turn are mainly determined by a small number (compared with $M-1$) of low-frequency zeros (Lane, 1988, §3.4) (i.e. those zeros close to the imaginary axis in the complex w-plane, whose mapping to the ζ-plane is described by (1.62)).

2.4.2 Multi-dimensional Case

Because of the non-uniqueness of the one-dimensional phase problem, indications that the two-dimensional phase problem may be unique (e.g. Napier and Bates, 1974; Baldwin and Warner, 1978; Fienup, 1978) seemed rather surprising when they first arose. Bruck and Sodin (1979) then argued that multi-dimensional phase problems can be expected to have unique solutions in all but very special circumstances, for the following reasons. Consider a two-dimensional discrete image $f[m,n]$, whose autocorrelation $a[m,n]$ has the z-transform $\mathcal{A}(\zeta,\gamma)$ (also known as the *autocorrelation polynomial*) given by (Oppenheim and Schafer, 1975)

$$\mathcal{A}(\zeta,\gamma) = \mathcal{F}(\zeta,\gamma)\,\mathcal{F}^*(1/\zeta^*, 1/\gamma^*) \tag{2.18}$$

Now there is no fundamental theorem of algebra in more than one dimension, meaning that multi-dimensional polynomials are *almost always* (i.e. always, except in special cases) *unfactorizable* or *irreducible* (Hayes and McClellan, 1982) (i.e cannot be factorized into the product of polynomials of lower order). Thus the autocorrelation polynomial $\mathcal{A}(\zeta,\gamma)$ can, because of its special nature, be factorized into the product of $\mathcal{F}(\zeta,\gamma)$ and $\mathcal{F}^*(1/\zeta^*, 1/\gamma^*)$, but no other factorizations are (in general) possible. The two-dimensional phase problem should thus be expected to (almost always) have a unique solution. The same argument, of course, applies to dimensions higher than two. Hayes (1982) has formalized the argument of Bruck and Sodin (1979) somewhat.

The zero-based arguments of §2.4.1 can be extended to the multi-dimensional case, in the following manner due to Lane *et al.* (1987). They show that a K-dimensional discrete image $f[\vec{m}]$, whose z-transform is $\mathcal{F}(\vec{\zeta})$, is zero on a continuous surface (or surfaces) in $2K$-dimensional z-space. This surface, termed the *zero-sheet* of $\mathcal{F}(\vec{\zeta})$, and denoted by $\mathcal{Z}\{\mathcal{F}(\vec{\zeta})\}$, is of dimension $2K-2$ (Lane *et al.*, 1987; Lane, 1988,

§4.5). It is now convenient to define $f'[m]$ as

$$f'[\vec{m}] = f^*[-\vec{m}] \tag{2.19}$$

so that the autocorrelation $a[\vec{m}]$ of $f[\vec{m}]$ becomes

$$a[\vec{m}] = f[\vec{m}] \odot f'[\vec{m}] \tag{2.20}$$

The z-transform of (2.20) yields

$$\mathcal{A}(\vec{\zeta}) = \mathcal{F}(\vec{\zeta})\,\mathcal{F}'(\vec{\zeta}) \tag{2.21}$$

from which it is evident that $\mathcal{A}(\vec{\zeta})$ is zero wherever $\mathcal{F}(\vec{\zeta})$ or $\mathcal{F}'(\vec{\zeta})$ is zero. The zero-sheet of $\mathcal{A}(\vec{\zeta})$ is thus the union of the zero-sheets of $\mathcal{F}(\vec{\zeta})$ and $\mathcal{F}'(\vec{\zeta})$, which is concisely expressed as

$$\mathcal{Z}\{\mathcal{A}(\vec{\zeta})\} = \mathcal{Z}\{\mathcal{F}(\vec{\zeta})\} \cup \mathcal{Z}\{\mathcal{F}'(\vec{\zeta})\} \tag{2.22}$$

For the one-dimensional case, the zero-sheet of $\mathcal{A}(\zeta)$ is of dimension $2K-2 = 0$, corresponding to the point zeros discussed in §2.4.1. For the two-dimensional case, however, the zero-sheet of $\mathcal{A}(\zeta,\gamma)$ consists of the union of the zero-sheets of $\mathcal{F}(\zeta,\gamma)$ and $\mathcal{F}^*(1/\zeta^*, 1/\gamma^*)$, each of which is a continuous two-dimensional surface in four-dimensional z-space. Because of the analytic nature of zero-sheets, it is possible to separate $\mathcal{Z}\{\mathcal{A}(\zeta,\gamma)\}$ into $\mathcal{Z}\{\mathcal{F}(\zeta,\gamma)\}$ and $\mathcal{Z}\{\mathcal{F}^*(1/\zeta^*, 1/\gamma^*)\}$. Note, however, that $\mathcal{Z}\{\mathcal{A}(\zeta,\gamma)\}$ cannot (in general) be separated in any other way. The concept of the zero-sheet thus demonstrates the important difference between the one-dimensional and the multi-dimensional phase problem.

A special situation in which there is more than one possible factorization of $\mathcal{A}(\zeta,\gamma)$ occurs when $f[m,n]$ is itself a convolution. Specifically, if

$$f[m,n] = f_1[m,n] \odot f_2[m,n] \tag{2.23}$$

then it follows that

$$\begin{aligned} \mathcal{A}(\zeta,\gamma) &= [\mathcal{F}_1(\zeta,\gamma)\,\mathcal{F}_2(\zeta,\gamma)]\,[\mathcal{F}_1^*(1/\zeta^*, 1/\gamma^*)\,\mathcal{F}_2^*(1/\zeta^*, 1/\gamma^*)] \\ &= [\mathcal{F}_1(\zeta,\gamma)\,\mathcal{F}_2^*(1/\zeta^*, 1/\gamma^*)]\,[\mathcal{F}_1^*(1/\zeta^*, 1/\gamma^*)\,\mathcal{F}_2(\zeta,\gamma)] \end{aligned} \tag{2.24}$$

Evidently, in this case $\mathcal{A}(\zeta,\gamma)$ can be factorized in two different ways, corresponding to two different image-forms, namely $f_1[m,n] \odot f_2[m,n]$ and $f_1[m,n] \odot f_2^*[-m,-n]$. In general, if a multi-dimensional image is a convolution of N components, then there are 2^N possible image-forms compatible with the given autocorrelation (Lane *et al.*, 1987).

It has been shown (Fiddy *et al.*, 1983; Nieto-Vesperinas and Dainty, 1985) that if a two-dimensional polynomial satisfies Eisenstein's criterion (which is equivalent to requiring that the support of the corresponding image has a special shape), then it is irreducible. More general constraints on support shapes which guarantee that any image with that support possesses an irreducible z-transform (and hence is uniquely related to its autocorrelation) are given by Crimmins (1987) and Brames (1987).

The discussion in both this and the previous sub-section has been restricted, for expositional convenience, to phase problems involving discrete images. It should be

noted, however, that the same conclusions have been arrived at for phase problems involving continuous images (Sanz and Huang, 1983; Barakat and Newsam, 1984). Furthermore, a proof of the uniqueness of solution of multi-dimensional phase problems, based on a physically meaningful interpolation argument (which leads directly to a practical algorithm) is given by Bates (1984).

2.4.3 Effect of Noise on Uniqueness

Since all measurements are unavoidably contaminated, for the reasons outlined in §1.1, it is important to question whether or not the conclusions of §2.4.1 and §2.4.2 are affected by the data being contaminated. Considering first the one-dimensional case, the effect of contamination of the autocorrelation a[m] (or, equivalently, contamination of the Fourier magnitude data) is to shift the positions of the zeros of $\mathcal{A}(\zeta)$ (Napier, 1971). Since the number of zeros remains unchanged from the noiseless case, so too does the number of possible solutions.

The effect of contamination upon the uniqueness of multi-dimensional phase problems is now discussed. Sanz *et al.* (1983) show that the uniqueness proof of Hayes (1982) is stable in the presence of noise on the input image. That is to say, if a multi-dimensional image has an irreducible z-transform, then an arbitrary (but small) perturbation of the input image also has an irreducible z-transform. If this was not so, then small perturbations of the input image may result in the phase problem having multiple solutions.

If the noiseless autocorrelation $a[m,n]$ is perturbed to $a_n[m,n]$, then its z-transform $\mathcal{A}_n(\zeta,\gamma)$ becomes irreducible (Sanz and Huang, 1985), implying that no image exists whose autocorrelation is $a_n[m,n]$. Despite this indication that practical multi-dimensional phase problems have (in theory) no solution, it seems intuitively reasonable that an image exists whose autocorrelation is close (in some sense) to $a_n[m,n]$. Numerous authors, such as Feldkamp and Fienup (1980) and Bates and Fright (1984), have shown empirically that this is so, a point which is discussed further in §4.1.

2.5 Examples of Phase Problems

To give some idea of the potential areas of application of the phase retrieval techniques described in subsequent chapters, it is now appropriate to describe a number of common phase problems. The oldest phase problem is that which occurs in X-ray crystallography, and so this is described first, in §2.5.1. Another significant phase problem is that which occurs in electron microscopy, as it gave rise to an iterative algorithm which is the basis of that invoked in Chapters 4 and 5, and so is described in §2.5.2. Astronomical speckle interferometry is described in §2.5.3, a detailed introduction to the astronomical seeing problem being given in §2.5.3.1 (which is also of relevance to the shift-and-add algorithm described in §2.8.2.1), with Labeyrie's speckle processing method, which gives rise to a phase problem, described in §2.5.3.2. A few further phase problems are introduced collectively in §2.5.4.

2.5.1 X-ray Crystallography

The field now known as X-ray crystallography was born in 1912 when Friedrich, Knipping and von Laue demonstrated the scattering of X-rays by a crystal. The mechanics of such scattering is now described, following the lines of Lipson and Taylor (1958). The density of electrons (which account for virtually all of the X-ray scattering) in a specimen (not necessarily a crystal) is denoted by $\rho(\vec{r})$, where $\vec{r}$ is an arbitrary position vector in a space known as *direct space* or *real space*. A parallel beam of monochromatic X-rays, termed the *incident wave*, impinges upon the specimen from a direction $\vec{s}_i$, and the *scattered wave* emanating from the specimen in an arbitrary direction $\vec{s}_s$ is observed.

To reduce the number of variables involved, it is customary to discuss scattering as a function of the scaled vector difference between $\vec{s}_i$ and $\vec{s}_s$ as

$$\vec{r}' = \frac{1}{\lambda}(\vec{s}_s - \vec{s}_i) \tag{2.25}$$

which is here termed the *normalized scattering direction*, where λ is the wavelength of the radiation, and both $\vec{s}_i$ and $\vec{s}_s$ are of unit length. It transpires (Lipson and Taylor, 1958, ch. 2) that the scattered wave function (relative to that scattered by a single electron at the origin) depends upon the normalized scattering direction $\vec{r}'$, and is given by

$$F(\vec{r}') = \int \rho(\vec{r}) \exp(i2\pi\vec{r} \cdot \vec{r}')\, d\vec{r} \tag{2.26}$$

Comparison of (2.26) with (1.14) reveals that $\rho(\vec{r})$ and $F(\vec{r}')$ constitute a Fourier transform pair. It should be pointed out that the dimensions of $\|\vec{r}'\|$ are the reciprocal of a length, because of the factor $1/\lambda$ in (2.25). This factor is included to ensure that $F(\vec{r}')$ is independent of λ, since $\vec{s}_s$ varies with λ (Lipson and Taylor, 1958, ch. 1). The vector $\vec{r}'$ is thus termed the *reciprocal vector*, and the space spanned by $\vec{r}'$ is known as *reciprocal space.*

A crystal consists of repetitions of a basic atomic structure, known as the *unit cell*, throughout the three-dimensional real space. The set comprising the points at the centre of each unit cell is termed the set of *lattice points*, and the three vectors which define three sides of the unit cell, and thereby span real space (but need not be mutually orthogonal) are denoted by $\vec{a}$, $\vec{b}$ and $\vec{c}$. It can be shown (Lipson and Taylor, 1958, ch. 3) that the scattered wave function $F(\vec{r}')$ can only possess significant value when $\vec{r}'$ satisfies

$$\begin{aligned} \vec{a} \cdot \vec{r}' &= h \\ \vec{b} \cdot \vec{r}' &= k \\ \vec{c} \cdot \vec{r}' &= l \end{aligned} \tag{2.27}$$

where h, k and l are all integers. The first condition in (2.28) constrains $\vec{r}'$ to lie upon a set of equispaced parallel planes, each plane being perpendicular to $\vec{a}$, and spaced by $1/\|\vec{a}\|$. When all three conditions in (2.28) are satisfied simultaneously, it is evident that $F(\vec{r}')$ can only assume significant value at a set of discrete points, known as *reciprocal lattice points*. This is, of course, entirely as should be expected, since

the spectrum of a periodic image exists only at the Nyquist points, as is explained in §1.5.

Since $\vec{r}'$ can be expressed in terms of h, k and l at the reciprocal lattice points by (2.28), and is essentially zero elsewhere, $F(\vec{r}')$ is usually written as $F(h,k,l)$. Because of the frequency of X-ray radiation ($\approx 10^{18}$Hz), it is only possible to measure $|F(h,k,l)|^2$. The making of such measurements is by no means a trivial task, since the data exist in a three-dimensional space (Buerger, 1944). The recovery of the electron density of the unit cell from $|F(h,k,l)|^2$ thus poses a phase problem. However, it is important to note that this is not a Fourier phase problem (§2.1), since $F(\vec{r}')$ only exists at the Nyquist points, and so it is clearly impossible to obtain samples with spacings satisfying (2.3).

The Fourier transform of $|F(\vec{r}')|^2$ yields the autocorrelation (§1.4) of $\rho(\vec{r})$, which is known as the *Patterson function*. The periodicity of the Patterson function is the same as that of $\rho(\vec{r})$, And hence consists of overlapped versions of the autocorrelation of the unit cell (Ramachandran and Srinivasan, 1970, ch. 2). For all but the simplest of crystals, the Patterson function contains so many overlapped atoms as to make its interpretation extremely difficult.

The crystallographic phase problem possesses significant *a priori* information, which is often sufficient to allow its solution. The electron density $\rho(\vec{r})$ is known to be positive and highly localized, and the number of atoms (and their atomic weights) are also known. Furthermore, numerous techniques exist for obtaining extra information, such as isomorphous replacement and anomalous scattering (Ramachandran and Srinivasan, 1970, chs. 9–13). The manner in which such *a priori* information can be utilized to help solve the crystallographic phase problem is mentioned in §3.2.1.

2.5.2 Microscopy

Conventional microscopes employing visible light are limited in resolution to about 200nm, due to the effects of diffraction (Hawkes, 1972, §1.1). The inherent resolution of electron microscopes, on the other hand, is significantly superior to that of light microscopes, because the de Broglie wavelength of electrons is extremely small ($\approx$ 4pm for 100kV electrons) compared to the wavelength of visible light ($\approx$500nm). However, aberrations in the electromagnetic lenses of electron microscopes limit the resolution of such devices to about 200 pm (i.e. 2Å) at present (Cowley and Smith, 1987). Hawkes (1972) provides a detailed introduction to electron microscopes and electron microscopy.

The basic operation of a typical transmission electron microscope is now briefly described. A beam of electrons, produced by an electron gun, passes through electromagnetic condenser lenses before encountering the specimen, located in the *object plane*. The electrons are scattered due to their interaction with the atoms of the specimen, and this scattering can be described in terms of changes to both (in general) the magnitude and phase of an electron wave as it propagates through the specimen. Electrons then pass through the objective lens, which contains a small hole (typically of $\approx 20 - 50\mu$m diameter) located in a plane known alternatively as the *diffraction plane, aperture plane*, or *back focal plane*. After passing through further intermediate and projection lenses, the electrons finally encounter a fluorescent

screen, photographic plate, or electronic camera, thereby producing an image known as an *electron micrograph.*

The form of the image thus obtained depends upon the particular configuration of the intermediate lenses, which can be straightforwardly modified, since the lenses are electromagnetic. When the microscope is configured in the *imaging mode*, a point in the object plane produces a corresponding point in the image, in which case the image is referred to as the *image plane micrograph.* Conversely, when the microscope is configured in the *diffraction mode*, a point in the diffraction plane produces a corresponding point in the image, in which case the image is referred to as the *diffraction plane micrograph.*

On denoting the wave function in the image plane by $f(x,y)$, it is evident that only $|f(x,y)|^2$ can be recovered directly from the image plane micrograph. Although $|f(x,y)|^2$ does indeed contain much useful information about the specimen, so too does $\mathrm{Ph}\{f(x,y)\}$, and thus a phase problem is seen to exist. However, the diffraction plane wave function (denoted here by $F(u,v)$) and the image plane wave function comprise a Fourier transform pair (Hawkes, 1972, §3.4; Saxton, 1978, §1.1), with $|F(u,v)|^2$ obtainable directly (after some trivial coordinate scaling) from the diffraction plane micrograph. The recovery of $\mathrm{Ph}\{f(x,y)\}$ (or, equivalently, $\mathrm{Ph}\{F(u,v)\}$), given both $|f(x,y)|^2$ and $|F(u,v)|^2$, is here referred to as the *electron microscopal phase problem* (often termed the *image/diffraction plane problem* (Saxton, 1978, ch. 4)), and is seen to be a phase problem of the kind defined in §2.1, but with additional *a priori* information.

A practical method exists for solving the electron microscopal phase problem, which is mentioned in §3.1.1.1. It is worth noting that it is possible to record other pairs of micrographs besides the image/diffraction plane pairs, such as defocus pairs, and one-sided diffraction plane pairs.

2.5.3 Astronomical Speckle Interferometry

From the time of Newton up until about 20 years ago, it had been accepted that the resolving power of optical astronomical telescopes was limited by the effects of inhomogeneities in the earth's atmosphere. To use astronomical parlance, there exists a *seeing problem* due to the atmosphere. The nature of the astronomical seeing problem is discussed in detail in §2.5.3.1.

In 1970, Labeyrie (1970) demonstrated that the seeing problem can be rather straightforwardly (but nonetheless ingeniously) overcome, thereby increasing the resolving power of large optical telescopes by perhaps a few orders of magnitude, and giving rise to a discipline which is now known as *astronomical speckle imaging* (Bates, 1982a). Labeyrie's method, called *astronomical speckle interferometry*, which provides an estimate of the Fourier transform magnitude of the object (and hence poses a phase problem) is described in §2.5.3.2. Dainty (1975, ch. 7) gives a concise introduction to astronomical speckle interferometry; more detailed reviews are provided by Bates (1982a) and Roddier (1988).

2.5.3.1 The Astronomical Seeing Problem

Optical telescopes are employed to form images of those celestial objects which either radiate (e.g. stars, comets) or reflect (e.g. planets, asteroids) radiation at optical frequencies. The conventional operation of a telescope involves tracking the object of interest across the sky, and integrating the light falling upon the focal plane of the telescope, with either photographic film or an electronic camera. Due to the enormous distance between an earth-based telescope and even the nearest celestial object, a Fourier transform relationship (§1.3) exists between the object distribution and its corresponding radiation pattern impinging upon the telescope. However, since the aperture of any telescope is necessarily of finite size, it can only intercept a portion of the aforesaid radiation pattern. For a telescope possessing a circular aperture of diameter D, the highest angular spatial frequency transmitted through the aperture (known as the *diffraction limit*) is D/λ cycles/radian, where λ is the wavelength of the radiation. The image formed in the focal plane of such a telescope is termed a *diffraction limited image*. Features of the object with spatial frequencies greater than the diffraction limit are said to be *unresolvable*.

The image obtained when a telescope possessing a circular aperture is directed upon an unresolvable object is known as the *Airy pattern* (Goodman, 1968, §4.2). The central lobe of the Airy pattern, called the *Airy disc*, has an angular diameter of approximately λ/D radians. Hence one would expect to be able to resolve two point sources separated by about λ/D radians, since the Airy discs corresponding to each source should then be sufficiently separated. As an example, a 5m telescope should be able to resolve a 0.02arc-sec double star. In practice, however, the image that one would obtain when observing such an object is typically of about 1arc-sec diameter when operating a telescope of such size in the conventional manner; the resolution is seen to be about 50 times worse than expected. The image obtained in such a situation is known as the *seeing disc*, which is roughly Gaussian in shape.

The reason for obtaining the seeing disc, as opposed to the Airy disc, is because the wave motion is distorted as it propagates through the earth's atmosphere, as a result of inhomogeneities (due mainly to temperature fluctuations) in the atmosphere's refractive index (Roddier, 1981). By writing the effective diameter of the seeing disc as λ/d, it is evident that the seeing disc could equivalently be obtained from an aperture of diameter d, where d is typically about 0.1m. It should be noted that d is equivalent to Fried's seeing parameter r_0 (Fried, 1966; Roddier, 1981).

It transpires that the atmospheric inhomogeneities can be modelled simply, but nonetheless usefully, as a conglomeration of *blobs* or *seeing cells* (Bates, 1982a, §§2.1,2.4). Each seeing cell has a diameter of roughly d, and an approximately uniform phase delay across the cell, but the relative phase delay between adjacent seeing cells is effectively random. A short exposure ($\approx$10ms) narrow band ($\approx$10nm) image of an unresolvable object recorded with a large telescope exhibits a granular appearance, known as *speckle* due to its similarity to laser speckle (Dainty, 1975). Such an image, called a *speckle image* or *specklegram*, results from interference between wavefronts passing through different seeing cells. The speckles are spread over an area about the size of the seeing disc, and (for an unresolvable object) the diameters of the speckles are roughly the size of the telescope's Airy disc.

If all the rays emanating from an extended object which impinge upon a particular point in the aperture plane of the telescope have passed through the same seeing cell, then the blurring of the image is isoplanatic (§1.4), and hence can be described by a (time varying) psf $h(x,y,t)$, termed the *speckle psf*. In practice, the blurring is effectively isoplanatic over a limited angular region of the celestial sphere, called the *isoplanatic patch*, which is seldom larger than 10arc-sec in diameter. For an object of smaller diameter than that of the isoplanatic patch, the *instantaneous speckle image* $s(x,y,t)$ is simply

$$s(x,y,t) = f(x,y) \odot h(x,y,t) + c(x,y,t) \tag{2.28}$$

where $f(x,y)$ is the ideal (diffraction limited) image, and the contamination $c(x,y,t)$ includes all departures of $s(x,y,t)$ from the simple convolutional model. The time dependence of $h(x,y,t)$ can be characterized by two parameters, namely the *stationary time* τ_s, which is the longest interval throughout which changes in $h(x,y,t)$ are negligible (typically $\approx$10ms), and the *redistribution time* τ_r, being the shortest interval within which $h(x,y,t)$ and $h(x,y,t+\tau_r)$ are effectively statistically independent (typically $\approx$500ms).

When a telescope is operated conventionally, the *long-exposure image* $f_{\rm le}(x,y)$ (i.e the seeing disc) is given by

$$\begin{aligned} f_{\rm le}(x,y) &= \overline{s(x,y,t)} \\ &= f(x,y) \odot \overline{h(x,y,t)} + \overline{c(x,y,t)} \end{aligned} \tag{2.29}$$

where the overbar denotes time averaging (see glossary). The form of $\overline{h(x,y,t)}$ is best described by considering its Fourier transform $\overline{H(u,v,t)}$. Only those spatial frequencies below about d/λ pass through the same seeing cell. The phase of $H(u,v,t)$ is thus essentially uncorrelated with the phase of $H(u,v,t+\tau_r)$ for spatial frequencies above about d/λ. So $\overline{H(u,v,t)}$, and hence also $\overline{S(u,v,t)}$, is effectively zero for spatial frequencies above about d/λ. The next section describes a procedure for processing speckle images which retains information up to the diffraction limit of the telescope.

2.5.3.2 Labeyrie's Speckle Processing Procedure

Antoine Labeyrie (1970) realized that, despite the fact that $\overline{s(x,y,t)}$ has little spatial frequency content above about d/λ, $s(x,y,t)$ possesses significant spatial frequency content up to the diffraction limit (D/λ) of the telescope. To retain spatial frequency information up to the diffraction limit, Labeyrie proposed averaging the Fourier intensities of the speckle images, rather than averaging the speckle images themselves (or, equivalently, their spectra), as occurs in conventional astronomical imaging. To do so, individual speckle images must first be recorded, before being subjected to further processing. The choice of the interval between, and the duration of, each exposure is based on the estimated values of τ_r and τ_s respectively, but is usually compromised by the desire to maximize the total number of detected photons.

It is now convenient to define the $m^{\rm th}$ member of an ensemble of M speckle images as $s_m(x,y)$, given by analogy with (2.28) as

$$s_m(x,y) = f(x,y) \odot h_m(x,y) + c_m(x,y) \tag{2.30}$$

By averaging the spectral intensity of each member of the ensemble of speckle images, one obtains

$$<|S_m(u,v)|^2> = |F(u,v)|^2 <|H_m(u,v)|^2> + C_c(u,v) \tag{2.31}$$

where the angular brackets denote ensemble averaging (see glossary), and $C_c(u,v)$ is the composite contamination, comprising $<|C_m(u,v)|^2>$ and two cross terms. Note that $<|H_m(u,v)|^2>$ has significant value up to the diffraction limit D/λ, and hence so too does $<|S_m(u,v)|^2>$. Before an estimate of $|F(u,v)|^2$ can be obtained, one must first estimate $<|H_m(u,v)|^2>$, which can be generated after observing an unresolvable object, known as a *reference object*, and performing the same processing on the speckle images thus obtained as that performed on the speckle images of the object of interest. To distinguish between quantities pertaining to the observation of the object of interest rather than the reference object, the variables of the latter are obtained by adorning those of the former with an r. It is then evident that

$$s_m^{\mathrm{r}}(x,y) = h_m^{\mathrm{r}}(x,y) + c_m^{\mathrm{r}}(x,y) \tag{2.32}$$

since the reference object is (by definition) unresolvable. The averaged speckle image spectral intensity for the reference object is then

$$<|S_m^{\mathrm{r}}(u,v)|^2> = <|H_m^{\mathrm{r}}(u,v)|^2> + C_c^{\mathrm{r}}(u,v) \tag{2.33}$$

If the seeing conditions during which the reference object is observed are statistically similar to those existing whilst the object of interest is observed, then $<|H_m(u,v)|^2>$ can be usefully approximated by $<|S_m^{\mathrm{r}}(u,v)|^2>$. An estimate of $|F(u,v)|^2$, denoted by $|\widehat{F(u,v)}|^2$, can then be formed by

$$|\widehat{F(u,v)}|^2 = <|S_m(u,v)|^2> \odot^{-1} <|S_m^{\mathrm{r}}(u,v)|^2> \tag{2.34}$$

where $\odot^{-1}$ denotes deconvolution, which is typically effected by Wiener filtering (see §3.3.1) or cleaning (Högbom, 1974; Bates and McDonnell, 1986, §17). By doing so, the effects of telescope aberrations are also removed, provided that the atmospheric fluctuations are more severe than the aberrations (Dainty, 1973; Bates, 1982a, §§4.2,8.9).

The processing described in this section, in which an estimate of the Fourier intensity (or, equivalently, the autocorrelation) of an astronomical object is obtained, is now known as *astronomical* (or *stellar*) *speckle interferometry*. Note that the recovery of the image $f(x,y)$ of the object from its estimated Fourier intensity $|\widehat{F(u,v)}|^2$ constitutes a phase problem.

2.5.4 Some Further Phase Problems

Three further practical phase problems are briefly introduced in this section, beginning with that which occurs in radio astronomy. Because of the frequency of radio astronomical sources of interest (typically 100MHz–100GHz), a radio telescope of enormous diameter is required to match the resolution of even a small optical telescope. Since construction of radio telescopes of such size is impossible, radio astronomers have devised a technique whereby a telescope of large effective diameter is

synthesized from two or more telescopes of much smaller diameter, a method known as *aperture synthesis* (Christiansen and Högbom, 1985, ch. 7).

On denoting the sky radio brightness distribution by $b(x,y)$, it can be shown that $B(u,v)$ can be obtained for a particular (u,v) from two radio telescopes configured as a correlation interferometer (Christiansen and Högbom, 1985, chs. 5,7). By varying the juxtaposition of the telescope pair (or, alternatively, employing other telescope pairs), samples of $B(u,v)$ can be obtained throughout the (u,v) plane, but note that, due to practical considerations, the samples seldom lie on a rectangular grid in the (u,v) plane. However, accurate measurement of $\mathrm{Ph}\{B(u,v)\}$ requires maintaining precise phase stability throughout the entire system, which is often difficult to achieve. The recovery of $b(x,y)$ from $|B(u,v)|^2$, when measurements of $\mathrm{Ph}\{B(u,v)\}$ are either poor or non-existent, poses a phase problem. Since astronomical radio sources are incoherent, one can make use of the *a priori* knowledge that $b(x,y)$ must be positive when attempting its reconstruction.

A further phase problem, in the general field of radio engineering, is that which occurs during the diagnosis of misalignments of large earth based satellite antennas (Gardenier *et al.*, 1986b). The type and degree of misalignments of such an antenna can be straightforwardly inferred from the phase of the antenna's aperture distribution function, denoted here by $a(x,y)$. Unfortunately, it is technically difficult to measure $|a(x,y)|^2$, and even more so to measure $\mathrm{Ph}\{a(x,y)\}$. However, the aperture distribution and the far field radiation pattern (denoted by $A(u,v)$) of the antenna comprise a Fourier transform pair (Silver, 1965, §6.3), and it so happens that $|A(u,v)|^2$ can be conveniently measured with the aid of a geostationary satellite. Recovery of $\mathrm{Ph}\{A(u,v)\}$ (or, equivalently, recovery of $a(x,y)$) from $|A(u,v)|^2$ then poses a phase problem. Fortunately, this particular phase problem possesses the following valuable *a priori* information. Firstly, $a(x,y)$ is of finite extent, and samples of $|A(u,v)|^2$ can be recorded sufficiently closely to satisfy (2.3). Secondly, an estimate of $|a(x,y)|^2$ can be obtained from the design data of the antenna under consideration. A detailed study by Gardenier (1990) of this phase problem indicates that useful solutions can be found from data typical of that obtainable in practice.

The final phase problem described here occurs when one wishes to determine the phase across an optical wavefront, a subject known as *wavefront sensing* (Bareket and Koliopoulos, 1982). A typical wavefront sensing situation arises when an optical wavefront, described by a pure phase function $\Theta(x,y)$, is incident upon an aperture with transmittance function $t(x,y)$. The resulting far field diffraction pattern $F(u,v)$ is detected, yielding a measurement of $|F(u,v)|^2$. The problem is then to recover $f(x,y)$, where

$$f(x,y) = t(x,y)e^{i\Theta(x,y)} \tag{2.35}$$

from knowledge of $|F(u,v)|^2$ and $a(x,y)$. Since $a(x,y)$ is typically either zero or unity, depending respectively upon whether (x,y) is outside or inside the aperture, it follows that $|f(x,y)|^2$ is effectively known. Thus the phase problem in wavefront sensing is essentially equivalent to that which occurs in electron microscopy (§2.5.2) (Gonsalves, 1982).

2.6 The Blind Deconvolution Problem

The convolution integral arises frequently in numerous branches of science and engineering, because (as explained in detail in §1.4) the output of a linear point spread invariant can be described as the convolution of the input to the system with the system psf. Systems of particular relevance to this thesis are imaging instruments, many of which (e.g. telescopes, cameras) can usefully be modelled as being point spread invariant (i.e. the image produced in the focal plane of such an instrument is simply the convolution of the original image with the instrumental psf).

A general contaminated convolution of a signal $f(\vec{x})$ with a psf $h(\vec{x})$ is here denoted by $g(\vec{x})$, where

$$g(\vec{x}) = f(\vec{x}) \odot h(\vec{x}) + c(\vec{x}) \tag{2.36}$$

The contamination $c(\vec{x})$ can include both contamination of the sort mentioned in §1.1, and any departures of $g(\vec{x})$ from the simple convolutional model. The recovery of a signal from a convolution is known (not surprisingly) as *deconvolution* (Bates and McDonnell, 1986, ch. 3). In many practical circumstances, the psf is either known *a priori*, or can be straightforwardly estimated (Andrews and Hunt, 1977, ch. 5; Bates and McDonnell, 1986, §52). Such situations are here said to constitute a *standard deconvolution problem*. The standard deconvolution problem is now posed as:

> "Given $g(\vec{x})$ and an estimate $\hat{h}(\vec{x})$ of $h(\vec{x})$, recover an estimate $\hat{f}(\vec{x})$ of $f(\vec{x})$, subject to any constraints on $f(\vec{x})$".

The standard deconvolution problem has been the subject of a considerable amount of research, spanning many years (e.g. Sondhi, 1972; Andrews and Hunt, 1977; Bates and McDonnell, 1986, ch. 3). Common methods of solving standard deconvolution problems are described in §3.3.1. However, situations sometimes arise in which the psf is unknown, or cannot be estimated to a sufficient accuracy to successfully apply a standard deconvolution technique. The recovery of a signal from a convolution in such situations has become known as *blind deconvolution*, a term due to Stockham, Jr. *et al.* (1975). The *blind deconvolution problem* is here posed as:

> "Given $g(\vec{x})$, recover estimates $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ of $f(\vec{x})$ and $h(\vec{x})$ respectively, subject to any constraints on $f(\vec{x})$ and $h(\vec{x})$".

On defining $f'(\vec{x})$ and $h'(\vec{x})$ by

$$f'(\vec{x}) = \alpha f(\vec{x} - \vec{x}') \quad \text{and} \quad h'(\vec{x}) = \frac{1}{\alpha} h(\vec{x} + \vec{x}') \tag{2.37}$$

where α is a (possibly complex) nonzero constant, and $\vec{x}'$ is an arbitrary position vector, it is seen that

$$f'(\vec{x}) \odot h'(\vec{x}) = f(\vec{x}) \odot h(\vec{x}) \tag{2.38}$$

It is convenient to say that $f'(\vec{x})$ and $h'(\vec{x})$ comprise an *image-form pair* of $f(\vec{x})$ and $h(\vec{x})$. Note that when presented with a blind deconvolution problem as posed above, one can only hope to recover an image-form pair of $f(\vec{x})$ and $h(\vec{x})$, rather than recover $f(\vec{x})$ and $h(\vec{x})$ themselves.

It is worth pointing out that the Fourier phase problem is merely a special case of the blind deconvolution problem. This can be seen by setting $h(\vec{x}) = f^*(-\vec{x})$ in (2.36), in which case $g(\vec{x}) = ff(\vec{x})$ (i.e. the autocorrelation of $f(\vec{x})$). Thus techniques for the analysis and solution of the blind deconvolution problem can be applied to the Fourier phase problem. However, since the Fourier phase problem has been the subject of intensive research for far longer than has the blind deconvolution problem, many of the techniques already developed for the Fourier phase problem have been generalized to apply to the blind deconvolution problem.

2.7 Uniqueness of Blind Deconvolution Problems

An important consideration, when attempting to solve any blind deconvolution problem, is whether a solution exists, and if so, whether it is unique. This question is now examined for both one-dimensional and multi-dimensional blind deconvolution problems, and the effect on uniqueness of data contamination is also discussed. Due to the relationship (discussed in the previous section) between the blind deconvolution problem and the Fourier phase problem, the techniques used in §2.4 are now applied to the question of uniqueness of the blind deconvolution problem.

Consider first the discrete convolution $g[m]$ of components $f[m]$ and $h[m]$, of extent L_f and L_h pixels respectively. The zeros of the convolution are related to the zeros of its components by (Bates and McDonnell, 1986, §14; Bates *et al.*, 1976)

$$\mathcal{Z}\{\mathcal{G}(\zeta)\} = \mathcal{Z}\{\mathcal{F}(\zeta)\} \cup \mathcal{Z}\{\mathcal{H}(\zeta)\} \tag{2.39}$$

from which it is evident that solving the blind deconvolution problem is equivalent to partitioning $\mathcal{Z}\{\mathcal{G}(\zeta)\}$ into $\mathcal{Z}\{\mathcal{F}(\zeta)\}$ and $\mathcal{Z}\{\mathcal{H}(\zeta)\}$. If L_f and L_h are both known *a priori*, then the ambiguity associated with this partitioning (i.e the number of possible ways in which the partitioning can be performed), denoted by *amb*, is given by

$$amb = \frac{(L_f + L_h - 2)!}{(L_f - 1)!\,(L_h - 1)!} \tag{2.40}$$

Note that the ambiguity is increased if L_f and L_h are unknown, but is reduced if further *a priori* information regarding either $f[m]$ or $h[m]$ (such as being real or positive) is available. The number of possible solutions to the one-dimensional blind deconvolution problem is thus seen to be hopelessly large, except in very special circumstances.

The degree of uniqueness of multi-dimensional blind deconvolution problems is, however, markedly different from that for the one-dimensional case (as is so for the Fourier phase problem). The discrete convolution $g[m,n]$ of $f[m,n]$ and $h[m,n]$ is characterized by its zero-sheet $\mathcal{Z}\{\mathcal{G}(\zeta,\gamma)\}$, which (for the reasons outlined in §2.4.2) can only be separated (in general) in one possible manner, namely into $\mathcal{Z}\{\mathcal{F}(\zeta,\gamma)\}$ and $\mathcal{Z}\{\mathcal{H}(\zeta,\gamma)\}$. The two-dimensional (and higher dimensional) blind deconvolution problem thus possesses (in general) a unique solution. Note, however, that if $f[m,n]$ or $h[m,n]$ is itself a convolution, then multiple solutions to the blind deconvolution problem exist, since $\mathcal{G}(\zeta,\gamma)$ can then be factorized in more than one manner.

For the one-dimensional case, the number of possible solutions to the blind deconvolution problem does not change when the convolution data becomes contaminated. This follows from the same reasoning applied in §2.4.3 to show that the number of solutions to the one-dimensional Fourier phase problem is unchanged by contamination of the autocorrelation data. When a two-dimensional convolution $g[m,n]$ becomes contaminated, forming $g_n[m,n]$, $\mathcal{G}_n(\zeta,\gamma)$ is then irreducible. Hence two-dimensional blind deconvolution problems in which the data are contaminated (which, of course, includes all practical applications) possess *no* exact solution. However, it seems reasonable that an approximate solution should exist, the convolution of which is "close to " (i.e. differs from by an amount of the order of the estimated noise level) $g_n[m,n]$. The results of Ayers and Dainty (1988) and Davey *et al.* (1989), and the results presented in §6.3, show this to be so.

2.8 Examples of Blind Deconvolution Problems

Since convolution arises in such a large number of situations, it follows that numerous potential blind deconvolution problems must exist. However, it has only recently been shown (Lane *et al.*, 1987) that multi-dimensional blind deconvolution problems can, in theory, be expected to possess unique solutions, and, more recently still (Lane and Bates, 1987b; Ayers and Dainty, 1988; Davey *et al.*, 1989), that implementable techniques exist for solving such problems, even when the data are appreciably contaminated. As soon as the applied science/engineering community grasps the significance of these discoveries, one can expect a multitude of applications to appear for such techniques.

Two significant well-known blind deconvolution problems are now described, namely deblurring of photographic images (§2.8.1), and deghosting of shift-and-add images (§2.8.2).

2.8.1 Photographic Deblurring

Two commonly encountered types of photographic blurring, namely motion and out-of-focus blurring, which can both be adequately modelled as convolutions, are now described. Note that this applies to images recorded with any form of camera, such as those using conventional photographic film, or CCD arrays.

The first type of photographic blurring to be described occurs when the camera (or, equivalently, the object) moves whilst the exposure is being made, and is known as *motion blur*. The simplest form of motion blur, called *uniform linear motion blur*, is that resulting from movement in a straight line, at constant speed. The psf in such cases is simply a straight line of uniform amplitude (Bates and McDonnell, 1986, §3; Andrews and Hunt, 1977, §4.4), which manifests itself in the spectrum of the convolution as lines of zero magnitude. From examination of the spacing and orientation of such lines, one can directly estimate the parameters (i.e. extent and orientation) of the psf (Bates and McDonnell, 1986, §52; Andrews and Hunt, 1977, §5.1), a technique known as *zero recognition*. The problem of restoration of an image degraded by uniform linear motion blur is thereby converted into a standard

deconvolution problem.

However, whilst estimation of the psf is straightforward for uniform linear motion blur, it is less so for more general motion blur. If the motion is linear (but not necessarily of uniform speed), then edge recognition (Bates and McDonnell, 1986, §52; Andrews and Hunt, 1977, §5.1), can often be employed to estimate the psf. For more complicated motion blurs, it is often impossible to adequately estimate the psf, thereby presenting a true blind deconvolution problem.

The other common form of photographic blurring occurs when a photograph is taken with an improperly focused camera. Such *out-of-focus blurring* is characterized by a psf which is (ideally) a circular disk (Bates and McDonnell, 1986, §3; Andrews and Hunt, 1977, §4.4), the diameter of which increases with the degree of defocus. Zero recognition can be applied to estimate the diameter of the psf in such circumstances (Bates and McDonnell, 1986, §52; Andrews and Hunt, 1977, §5.1), and restored images obtained by deconvolution with an assumed disk-shaped psf are often adequate. In practice, however, the shape of the psf of an out-of-focus camera is determined by the design of the iris of the camera in question, and often may be approximately pentagonal or hexagonal in shape, rather than circular.

It should be pointed out that any part of the blurred image which falls outside the recording medium of the camera is irretrievably lost. If any non-zero parts of the blurred image are lost in this manner then the remaining part of the blurred image is *not* (in general) a convolution, and appropriate pre-processing (Bates and McDonnell, 1986, §15) must be performed before deconvolution is attempted. Furthermore, since recording devices usually possess non-linear response characteristics (e.g. photographic film, see Andrews and Hunt, 1977, §2.4), pre-processing is required to take such characteristics into account, otherwise the blurred image does not even approximate a convolution.

2.8.2 Deghosting of Shift-and-add Images

As described in §2.5.3, astronomical speckle interferometry is capable of providing diffraction limited estimates of the autocorrelation of astronomical objects, despite the effects of the earth's turbulent atmosphere. A computationally straightforward astronomical speckle imaging technique known as shift-and-add (Bates, 1982a, §8.7), which is capable of providing diffraction limited estimates of such objects (not merely their autocorrelations) is described in detail in the following section. A shortcoming of the shift-and-add technique, however, is its propensity to produce artefacts in the image, known as "ghosts", as explained in §2.8.2.2. The recovery of the true image from a ghosted shift-and-add image is shown to pose a blind deconvolution problem.

2.8.2.1 The Shift-and-add Algorithm

Shift-and-add (Bates, 1982a, §8.7) is unarguably the most computationally undemanding speckle imaging algorithm presently available, and so has been the subject of much practical and theoretical attention. To understand the reasoning behind the shift-and-add algorithm, it is necessary to consider the detailed nature of a speckle image (§2.5.3.1). A typical speckle image of an object comprising a number of sepa-

rated parts (all within the same isoplanatic patch), each of which is unresolvable (e.g. a multiple star), consists of the sum of shifted and weighted versions of the speckle psf (i.e. the speckle image that would be obtained from a single unresolvable object). As explained in §2.5.3.1, a typical speckle psf consists of a number of individual speckles, each of which is roughly the size of the Airy disc of the telescope. Thus the speckle image of a multiple star can equivalently be looked upon as the sum of many versions of the ideal (diffraction limited) image, each version being scaled, shifted, and distorted according respectively to the brightness, position, and shape of each speckle of the speckle psf. It should be pointed out that these versions of the ideal image are usually so severely overlapped that the ideal image is not immediately recognizable in a typical speckle image. However, it is useful to postulate that the brightest parts of a speckle image are versions of the brightest part of the true image.

Lynds *et al.* (1976) first noted the points discussed in the previous paragraph, and devised an algorithm in which an image is generated by superimposing a number of the brightest speckles in a single speckle image. Unfortunately, the most interesting astronomical objects tend to be so faint that very few complete speckles are contained in a typical speckle image. Furthermore, it seems intuitively reasonable that a blurred image should be least distorted where it is brightest. These are some of the motivations behind Bates and Cady's (1980) aptly named *shift-and-add* algorithm (SAA), in which each speckle image is shifted, such that its brightest point lies at the centre of image space, and then added to the sum of the previously such shifted speckle images. The *shift-and-add image* $saa(x,y)$ thereby generated from an ensemble of M speckle images, the m^{th} of which is $s_m(x,y)$ (as defined in §2.5.3.1) is simply

$$\begin{aligned} saa(x,y) &= <s_m(x+x_m, y+y_m)> \\ &= f(x,y) \odot h_{\text{saa}}(x,y) + c_{\text{c}}(x,y) \end{aligned} \tag{2.41}$$

where (x_m, y_m) is the coordinate of the brightest point in $s_m(x,y)$, $h_{\text{saa}}(x,y)$ is the *shift-and-add psf* given by

$$h_{\text{saa}}(x,y) = <h_m(x+x_m, y+y_m)> \tag{2.42}$$

and $c_{\text{c}}(x,y)$ is the composite contamination (which, from hereon, is used to denote any – i.e. not necessarily the same – composite contamination). Because of the minimal computational requirements of the shift-and-add algorithm, it can easily be performed in real time (Cady, 1980), in contrast to other speckle imaging algorithms, thereby avoiding the need to store individual speckle images.

An example of the operation of the shift-and-add algorithm on computer-generated speckle data is given in Figure 2.2. The reader is referred to Davey (1989, §4.4) for a detailed description of computer simulation of speckle images. Note that the true image appears atop a background, known as a *fog*, which tends to become smoother as M (the number of speckle images from which the shift-and-add image is generated) increases. The fog results from the addition of the remaining randomly positioned distorted versions of the true image present in each speckle image, and is similar in nature to the seeing disc (§2.5.3.1).

To allow the fog to be removed from $saa(x,y)$, an estimate of the fog can be obtained by performing shift-and-add on an unresolvable object. Using the same

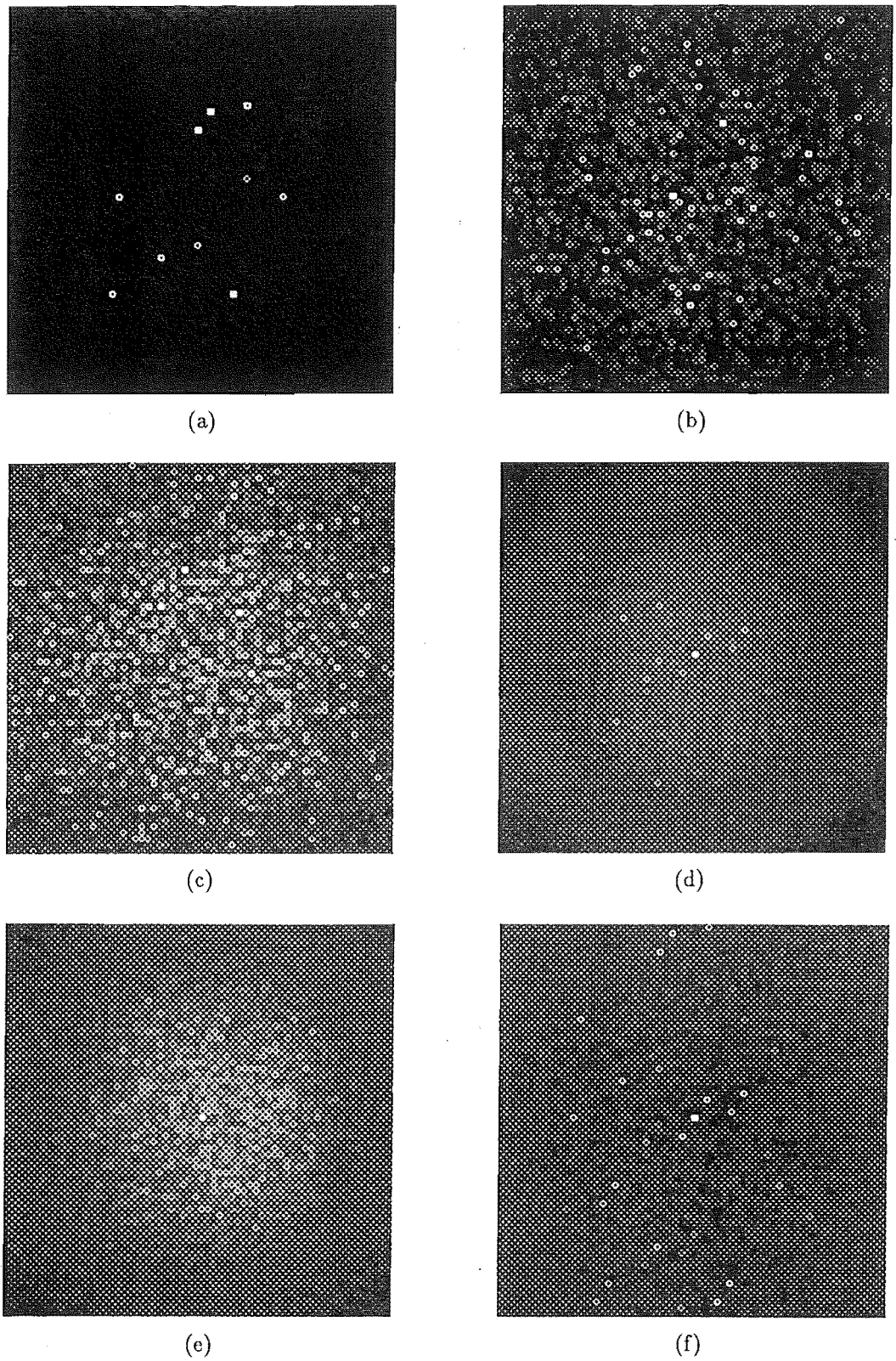


Figure 2.2 Example of performing shift-and-add processing on computer-simulated speckle data. (a) the original image $f(x, y)$, (b) a typical speckle psf $h_m(x, y)$, (c) a typical speckle image $s_m(x, y)$, (d) shift-and-add image $saa(x, y)$ computed from 200 speckle images, (e) shift-and-add image $saa^{\mathrm{r}}(x, y)$ of a reference object, and (f) defogged shift-and-add image $saa_{\mathrm{d}}(x, y)$ obtained by Wiener filtering (e) from (d).

variables describing shift-and-add performed on the object, but adorning them with an 'r' to indicate that they refer to observations of a reference object, it is evident that

$$\begin{aligned} saa^{\mathrm{r}}(x,y) &= <s^{\mathrm{r}}_m(x+x_m, y+y_m)> \\ &= h^{\mathrm{r}}_{\mathrm{saa}}(x,y) + c_{\mathrm{c}}(x,y) \end{aligned} \tag{2.43}$$

The *defogged shift-and-add image* $saa_{\mathrm{d}}(x,y)$ is then obtained from

$$saa_{\mathrm{d}}(x,y) = saa(x,y) \odot^{-1} saa^{\mathrm{r}}(x,y) \tag{2.44}$$

where the deconvolution is typically effected by Wiener filtering (§3.3.1) or cleaning (Bates and McDonnell, 1986, §17). For the class of objects which possess a dominatingly bright point, $h^{\mathrm{r}}_{\mathrm{saa}}(x,y)$ tends to be usefully similar to $h_{\mathrm{saa}}(x,y)$, in which case $saa_{\mathrm{d}}(x,y)$ is typically an acceptable estimate of the true (diffraction limited) image, as indicated by Figure 2.2. The next section discusses the behaviour of the shift-and-add algorithm when the object is not of this class.

2.8.2.2 Generation of Ghosts by Shift-and-add

If shift-and-add is performed on speckle images obtained from an object which does not possess a dominatingly bright point, then the brightest point in each speckle image does not necessarily correspond to the brightest point in the true image. Instead, the brightest point in a particular speckle image may correspond to any of the brighter points in the true image, and so the shift-and-add image contains multiple shifted and scaled versions of the true image. Such artefacts are known as *ghosts* (Bates and Cady, 1980), the generation of which has been described analytically for objects comprising distinct points (Bagnuolo, Jr., 1982; Hunt *et al.*, 1983) and for extended objects (Hunt *et al.*, 1983; Bagnuolo, Jr., 1985). Figure 2.2 demonstrates ghosting of the shift-and-add image generated from an object consisting of multiple parts of similar brightness.

From the above discussion, it is evident that the shift-and-add psf $h_{\mathrm{saa}}(x,y)$ is object dependent, and so cannot in general be approximated by $h^{\mathrm{r}}_{\mathrm{saa}}(x,y)$. It is thus convenient to partition $h_{\mathrm{saa}}(x,y)$ (Bates and Davey, 1987a) into one part which is object dependent, namely $h^{\mathrm{o}}_{\mathrm{saa}}(x,y)$, and another part which is object independent (i.e dependent only upon the seeing), namely $h^{\mathrm{s}}_{\mathrm{saa}}(x,y)$, so that

$$h_{\mathrm{saa}}(x,y) \approx h^{\mathrm{o}}_{\mathrm{saa}}(x,y) \odot h^{\mathrm{s}}_{\mathrm{saa}}(x,y) \tag{2.45}$$

Since $h^{\mathrm{r}}_{\mathrm{saa}}(x,y)$ is likely to be a useful estimate of $h^{\mathrm{o}}_{\mathrm{saa}}(x,y)$, it then follows from (2.41) (2.44) and (2.45) that the defogged shift-and-add image is related to the true image by

$$saa_{\mathrm{d}}(x,y) = f(x,y) \odot h^{\mathrm{o}}_{\mathrm{saa}}(x,y) + c_{\mathrm{c}}(x,y) \tag{2.46}$$

Since $h^{\mathrm{o}}_{\mathrm{saa}}(x,y)$ is unknown (because it is dependent upon $f(x,y)$), the recovery of $f(x,y)$ from $saa_{\mathrm{d}}(x,y)$ poses a blind deconvolution problem, of the form defined in §2.6. This particular blind deconvolution problem possesses significant *a priori* information, in that $h^{\mathrm{o}}_{\mathrm{saa}}(x,y)$ can be estimated for a given $f(x,y)$ (Bagnuolo, Jr., 1985).

Note in particular that an object consisting of isolated points results in $h^{\circ}_{\mathrm{saa}}(x,y)$ possessing points in the same positions as those in $f(-x,-y)$, hence $saa_{\mathrm{d}}(x,y)$ is similar to $ff(x,y)$. However, the ratio of the brightness of the corresponding points in $h^{\circ}_{\mathrm{saa}}(x,y)$ to those in $f(-x,-y)$ is higher for the brighter points in $f(-x,-y)$.

Various extensions to shift-and-add (referred to as *simple shift-and-add* for the remainder of this section, to avoid confusion with its extensions) have been proposed and demonstrated. The aim of such extensions is to eliminate the fog and/or ghosts present in simple shift-and-add images. *Correlation shift-and-add* (Bates *et al.*, 1985; Minard, 1985, §6.5), or its close relative *matched-filter shift-and-add* (Ribak *et al.*, 1985), attempts to overcome ghosting by correlating each speckle image with an estimate of the true image (e.g. an image produced by basic shift-and-add) and shifting each speckle image according to the position of the brightest point in this correlation. Although effective in minimizing ghosting, correlation shift-and-add is significantly more computationally demanding than basic shift-and-add, since a correlation must be performed on *each* speckle image. Combining basic shift-and-add with a blind deconvolution algorithm is a much more computationally attractive option (Bates and Davey, 1987b), since the computationally undemanding basic shift-and-add can be performed in real time on a simple computer (or simple dedicated hardware), the resulting shift-and-add image then being transferred to a more powerful computer to effect defogging and deghosting.

Finally, it should be pointed out that a further advantage of shift-and-add is that it is a type of first order interferometry (Bates and Gough, 1975). As such, it can be expected (Bates and Davey, 1987b) to be less sensitive to contamination than second order methods, such as speckle interferometry (§2.5.3) and Knox-Thompson processing (Knox and Thompson, 1974).

Chapter 3

Solution of Phase Problems and Blind Deconvolution Problems

This chapter reviews existing methods for solving the two image reconstruction problems which are the subject of this thesis, namely the phase problem and the blind deconvolution problem. General image reconstruction techniques, which can be usefully applied to both these problems, are presented in §3.1. Methods specific to the solution of the phase problem are described in §3.2, whilst methods particular to the solution of the blind deconvolution problem are described in §3.3.

3.1 General Image Reconstruction Techniques

General image reconstruction techniques, which are applicable to both the phase problem and the blind deconvolution problem, are grouped together in this section. Iterative image reconstruction techniques are discussed in §3.1.1, while minimization techniques are described §3.1.2. The precise details of how a particular technique is applied to either the phase or blind deconvolution problem are presented in the appropriate subsection of §3.2 and §3.3.

3.1.1 Iterative Techniques

Many image reconstruction problems can be solved with the aid of iterative techniques. The advantages of iterative techniques include robustness when operating on contaminated data, ease of incorporation of *a priori* information into the reconstruction algorithm, and the relative simplicity with which they can be implemented. However, iterative techniques tend to be computationally demanding in general, because of their repetitive nature.

A general iterative image reconstruction method, called the iterative transform algorithm, which is applicable in situations in which the *a priori* information (i.e. both measured data and constraints) is distributed between two different spaces (e.g. image and Fourier space) is described in §3.1.1.1. Another general image reconstruction method known as projections onto convex sets, which provides valuable insight into the behaviour of the iterative transform algorithm, is described in §3.1.1.2.

3.1.1.1 The Iterative Transform Algorithm

Many iterative image reconstruction algorithms are merely special cases of a general algorithm, which is here termed the *iterative transform algorithm*. The idea behind the iterative transform algorithm is to alternately transform an image estimate between image and Fourier space, imposing the appropriate constraints on the image estimate in each domain, in the hope that it eventually satisfies the constraints in both domains. The first image reconstruction algorithm of this form was that proposed by Gerchberg and Saxton (1972), although similar algorithms had been previously proposed in other areas (see Fienup, 1981b). Following publication of the Gerchberg-Saxton algorithm, a number of image reconstruction algorithms of this general class were reported, as Fienup (1981b) relates.

The iterative transform algorithm is depicted in Figure 3.1, where $\tilde{f}(\vec{x})$ and $\bar{f}(\vec{x})$ denote estimates of $f(\vec{x})$ (the original image) which satisfy the image and Fourier domain constraints respectively. The algorithm is here initiated with some starting image $f_0(\vec{x})$ for $\bar{f}(\vec{x})$, but could equivalently be started at any point in the loop. Next, $\bar{f}(\vec{x})$ is forced to meet the image domain constraints, thereby forming $\tilde{f}(\vec{x})$, which is then Fourier transformed, yielding $\tilde{F}(\vec{u})$. This $\tilde{F}(\vec{u})$ is then forced to meet the Fourier domain constraints, thereby forming $\bar{F}(\vec{u})$, which is then inverse Fourier transformed, yielding a new $\bar{f}(\vec{x})$. An iterative loop is thereby established. The loop is traversed repeatedly, until either a pre-determined number of iterations have been performed or $\bar{f}(\vec{x})$ and $\tilde{f}(\vec{x})$ become sufficiently close to one another (according to a pre-set criterion).

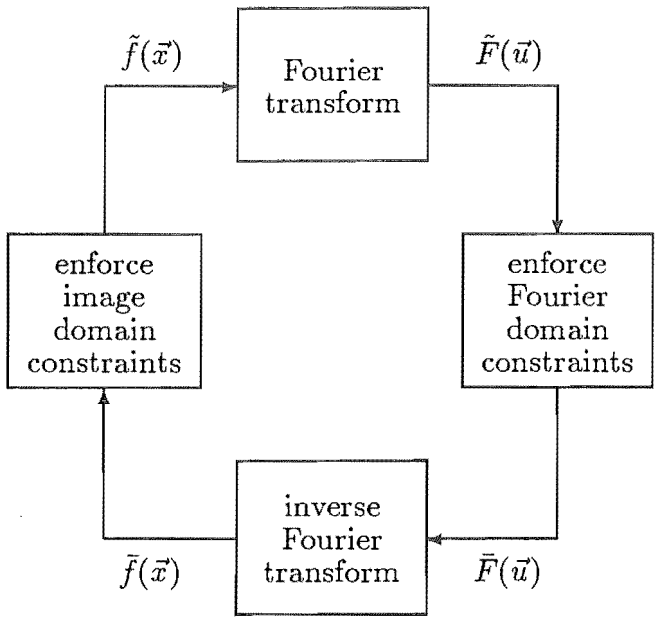


Figure 3.1 The basic loop of the iterative transform algorithm.

Mention is now made of the types of constraints which are commonly incorporated into the iterative transform algorithm, and how such constraints can be imposed. Note that the constraints can include measured data, as well as *a priori* information regarding the original image. The manner in which the constraints are applied is typically devised so as to make the minimum change to the quantity being

constrained (Fienup, 1981b). For example, the image space constraints are typically applied by selecting $\tilde{f}(\vec{x})$ to be that image which is closest (in some sense) to $\bar{f}(\vec{x})$ and which satisfies the constraints. Table 3.1 lists various constraints which can be incorporated into the iterative transform algorithm, and the typical method by which each constraint is imposed. The first two constraints listed in Table 3.1 result from *a priori* knowledge regarding the original image. Conversely, the last four constraints listed in Table 3.1 correspond to information regarding the original image that is obtained from measurements.

Constraint	Method of Imposition
image positivity i.e. $f(\vec{x}) \geq 0$	$\tilde{f}(\vec{x}) = \begin{cases} \bar{f}(\vec{x}) & \vec{x} : \bar{f}(\vec{x}) > 0 \\ 0 & \text{otherwise} \end{cases}$
image support i.e. $f(\vec{x}) = 0$ for $\vec{x} : \vec{x} \notin S_f(\vec{x})$	$\tilde{f}(\vec{x}) = \begin{cases} \bar{f}(\vec{x}) & \vec{x} : \bar{f}(\vec{x}) \in S_f(\vec{x}) \\ 0 & \text{otherwise} \end{cases}$
given image magnitude i.e. $\lvert f(\vec{x})\rvert$ known	$\tilde{f}(\vec{x}) = \lvert f(\vec{x})\rvert \exp(i\,\mathrm{Ph}\{\bar{f}(\vec{x})\})$
given Fourier magnitude i.e. $\lvert F(\vec{u})\rvert$ known	$\bar{F}(\vec{u}) = \lvert F(\vec{u})\rvert \exp(i\,\mathrm{Ph}\{\tilde{F}(\vec{u})\})$
given Fourier phase i.e. $\mathrm{Ph}\{F(\vec{u})\}$ known	$\bar{F}(\vec{u}) = \lvert \tilde{F}(\vec{u})\rvert \exp(i\,\mathrm{Ph}\{F(\vec{u})\})$
given limited spectrum i.e. $F(\vec{u})$ known for $\vec{u} : \vec{u} \in R_F$	$\bar{F}(\vec{u}) = \begin{cases} F(\vec{u}) & \vec{u} : \vec{u} \in R_F \\ \tilde{F}(\vec{u}) & \text{otherwise} \end{cases}$

Table 3.1 Common constraints applied in the iterative transform algorithm.

The majority of image reconstruction algorithms which can be described in terms of the iterative transform algorithm employ some combination of the constraints listed in Table 3.1. A number of such image reconstruction algorithms are listed in Table 3.2, together with the particular constraints imposed in each algorithm. Further details regarding the Gerchberg-Saxton algorithm are given by Gerchberg and Saxton (1972) and Saxton (1978, chs. 5,6), whilst the Gerchberg-Papoulis algorithm (often referred to as simply Gerchberg's algorithm) is described by Gerchberg (1974) and Papoulis (1975). A related super-resolution algorithm is described by Gerchberg (1989). Fienup's (1978) phase retrieval algorithm is discussed in greater detail in §3.2.8.1, whereas the Hayes-Lim-Oppenheim magnitude retrieval algorithm is described by Hayes *et al.* (1980) and Hayes (1982).

It is worth mentioning that the iterative transform algorithm can also be employed to solve image synthesis problems, as well as image reconstruction problems (Fienup, 1981b). Image synthesis problems (sometimes referred to as image design problems) involve the determination of an image which exhibits certain desired characteristics in both image and Fourier space. When the iterative transform algorithm is employed to solve image synthesis problems, the constraints applied in the algorithm correspond to these desired characteristics. Some examples of image synthesis

Algorithm name	Constraints applied
Gerchberg-Saxton phase retrieval algorithm	given image magnitude, given Fourier magnitude
Gerchberg-Papoulis super-resolution algorithm	given limited spectrum, image support
Fienup's phase retrieval algorithm	given Fourier magnitude, image support, image positivity
Hayes-Lim-Oppenheim magnitude retrieval algorithm	given Fourier phase, image support

Table 3.2 A selection of image reconstruction algorithms which can be described in terms of the iterative transform algorithm.

problems, in which the iterative transform algorithm has been applied, include the design of holographic (Galagher and Liu, 1973) and microphotographic (Saleh, 1987) masks.

3.1.1.2 Method of Projections onto Convex Sets

This section describes a general image reconstruction technique which is known as the *method of projections onto convex sets*, or simply the *method of convex projections*. This technique was first described by Youla (1978), and extended by Youla and Webb (1982). An extremely thorough treatment of the subject is presented by Youla (1987).

The original image f is considered to be a vector in a Hilbert space $\mathcal{H}$, a Hilbert space being a normed vector space possessing an inner product (Pryce, 1972, §12). Although the results stated here hold for any $\mathcal{H}$, some useful choices of $\mathcal{H}$ include (Pryce, 1972, §5) the complex L_2 space (i.e. the space of square-integrable functions of a single variable), with norm

$$\|f\| = \left[\int |f(x)|^2 dx\right]^{\frac{1}{2}}, \tag{3.1}$$

or the complex l_2 space (i.e. the space of square-summable functions of a single variable), with norm

$$\|f\| = \left[\sum_j |f_j|^2\right]^{\frac{1}{2}}, \tag{3.2}$$

or the corresponding spaces, $L_{2\times2}$ and $l_{2\times2}$, of functions of two variables.

Each constraint (i.e. measured data or *a priori* knowledge) placed upon the image f is regarded as constraining f to lie in a corresponding subset of $\mathcal{H}$. If f belongs to M such sets, denoted by $C_1, C_2, \ldots, C_M$, then it is evident that f must

belong to the intersection C_0 of these M sets. This is written as

$$f \in C_0 \quad \text{where} \quad C_0 = \bigcap_{m=1}^{M} C_m \tag{3.3}$$

The image reconstruction problem is thus reduced to the problem of finding any $\hat{f}$ such that $\hat{f} \in C_0$. If C_0 contains a single element then the image reconstruction problem has a unique solution, whereas if C_0 contains more than one element then multiple solutions exist. An empty C_0 means that no solution exists (i.e. the constraints are inconsistent).

It is now convenient to assume that each constraint set C_m is convex, which means that

$$\text{if} \quad a, b \in C_m \quad \text{then} \quad t\,a + (1-t)\,b \in C_m \quad \text{for} \quad 0 \leq t \leq 1 \tag{3.4}$$

Furthermore, the *projection operator* for the set C_m, denoted by P_m, is defined by

$$\|a - P_m a\| = \min_{b \in C_m} \|a - b\| \tag{3.5}$$

where the operator notation $P_m a$ is invoked to denote the result of P_m operating upon a. $P_m a$ is termed the *projection* of a onto C_m, and is simply the element of C_m which is closest to (i.e. shares the smallest norm with) a. It can be shown (Youla and Webb, 1982) that the sequence $\{\hat{f}_n\}$ obtained by the recursion

$$\hat{f}_{n+1} = P_M\, P_{M-1} \ldots P_1\, \hat{f}_n \tag{3.6}$$

converges weakly to a point in C_0, regardless of the choice of f_0. The weak convergence of $\{a_n\}$ to b is defined by $\lim_{n\to\infty} \langle a_n, c\rangle = \langle b, c\rangle \quad \forall c \in \mathcal{H}$, where $\langle\cdot\rangle$ denotes the inner product, and c is arbitrary. In addition, the recursion given by

$$\hat{f}_{n+1} = T_M\, T_{M-1} \ldots T_1\, \hat{f}_n \tag{3.7}$$

where T_m is the m^{th} *relaxed projection operator*, itself given by

$$T_m = (1 - \lambda_m) I + \lambda_m P_m \tag{3.8}$$

where λ_m is the m^{th} *relaxation parameter* and I is the identity operator, can also be shown to converge weakly. Youla and Webb (1982) derive conditions under which the recursion (3.7) converges strongly (the strong convergence of $\{a_n\}$ to b implies $\lim_{n\to\infty} \|a_n - b\| = 0$). One such condition is that $\mathcal{H}$ possesses finite dimension, which must necessarily be so when (3.7) is computed, since $\hat{f}$ must then be described by a finite number of samples. The use of relaxation parameters can often accelerate convergence of the algorithm (Sezan and Stark, 1982), especially if the relaxation parameters are adaptively optimized (Levi and Stark, 1987).

The theory of projections onto convex sets thus leads directly to implementable algorithms, for which convergence is guaranteed (provided all the constraints can be expressed as convex sets). All of the constraints listed in Table 3.1, except for those two involving given magnitudes, correspond to convex sets (Youla and

Webb, 1982). Furthermore, of those constraints listed in Table 3.1 which do correspond to convex sets (apart from given Fourier phase), the corresponding method of application of each constraint is equivalent to a projection (Youla and Webb, 1982). While Sezan and Stark (1983) point out that contamination of the data usually results in the solution set C_0 being empty, in which case the projections onto convex sets algorithm may diverge, they suggest various methods for overcoming this problem.

It is interesting to compare the typical convergence properties of the algorithms listed in Table 3.2. The Gerchberg-Papoulis super-resolution algorithm, which is equivalent to projections onto convex sets, usually converges (Gerchberg, 1974), although sometimes rather slowly (Papoulis, 1975). The Hayes-Lim-Oppenheim magnitude retrieval algorithm, for which the constraints are convex sets but the phase constraint is not applied as a projection, also usually converges, as does the algorithm of Levi and Stark (1983) which employs true projections. However, Fienup's error-reduction algorithm, and (to a lesser extent) the Gerchberg-Saxton algorithm, both of which include magnitude constraints (which do not correspond to convex sets), often exhibit convergence difficulties (Fienup, 1982; Gerchberg, 1986).

It is worth mentioning that the projections onto convex sets algorithm has been implemented in closed form for certain applications (Sabri and Steenaart, 1978; Marks and Smith, 1981). Furthermore, the extension of the theory of projections onto convex sets to image reconstruction problems involving non-convex sets is discussed in §3.2.8.4.

3.1.2 Optimization Techniques

This section explains how general optimization techniques may be applied to image reconstruction problems, and some appropriate techniques are introduced. Specific applications of optimization techniques to the phase problem and blind deconvolution problem are later described in §3.2.9 and §3.3.4 respectively.

Consider an imaging situation in which some data $d(\vec{w})$ is measured in a space which need not necessarily be the same as that in which the original image $f(\vec{x})$ exists. It is convenient to define a functional $T\{f(\vec{x})\}(\vec{w})$ which models the ideal (i.e. uncontaminated) data that would result from the image $f(\vec{x})$. The actual measured data $d(\vec{w})$ can thus be expressed as

$$d(\vec{w}) = T\{f(\vec{x})\}(\vec{w}) + c(\vec{w}) \tag{3.9}$$

where $c(\vec{w})$ is the contamination. The problem of finding an estimate $\hat{f}(\vec{x})$ of $f(\vec{x})$ then reduces to the problem of finding the image which minimizes the *cost function*, or *objective function*, $Q\{\hat{f}(\vec{x})\}$ as given by

$$Q\{\hat{f}(\vec{x})\} = \int \left[d(\vec{w}) - T\{\hat{f}(\vec{x})\}(\vec{w})\right]^2 d\vec{w} \tag{3.10}$$

where the integral is over all values of $\vec{w}$ for which $d(\vec{w})$ is significant. The image reconstruction problem is thereby reduced to a minimization problem. If constraints are placed upon $\hat{f}(\vec{x})$, the image reconstruction problem becomes a constrained minimization problem.

The ease with which the minimization problem associated with a particular image reconstruction problem can be solved depends upon the topology of $Q\{\hat{f}(\vec{x})\}$ for that problem. If $Q\{\hat{f}(\vec{x})\}$ possesses a single minimum, in which case $Q\{\hat{f}(\vec{x})\}$ is said to be *unimodal*, the minimization techniques described in §3.1.2.1 may be employed to find that minimum. However, if $Q\{\hat{f}(\vec{x})\}$ possesses multiple minima, in which case $Q\{\hat{f}(\vec{x})\}$ is said to be *multimodal*, techniques of the kind described in §3.1.2.2 must be employed. An alternative method of posing an image reconstruction problem as an optimization problem, which is especially useful when the data are appreciably contaminated, is discussed in §3.1.2.3.

3.1.2.1 Gradient Methods for Minimization

Numerous methods exist (e.g. Press *et al.*, 1986, ch. 10) for finding the minimum of a unimodal function $Q(\vec{x})$, where $\vec{x}$ is some N-dimensional vector. Methods which make use of $\nabla Q(\vec{x})$, the gradient of $Q(\vec{x})$, are commonly termed *gradient methods* (Polak, 1971, ch. 2). Such methods typically reach an adequate solution in significantly fewer steps than methods which do not utilize gradient information.

The simplest gradient method is the *steepest descent method* (Polak, 1971, §2.1), which operates by generating a succession of points, the k^{th} of which is denoted by $\vec{x}_k$. The point $\vec{x}_{k+1}$ is that point for which $Q(\vec{x})$ assumes the minimum value along a line from $\vec{x}_k$ in the direction $-\nabla Q(\vec{x})$. However, the steepest descent method often proves to be extremely slow in finding the minimum of $Q(\vec{x})$, as it tends to take a large number of small steps if $Q(\vec{x})$ possesses long, narrow valleys.

To circumvent the problem of slow convergence of the steepest descent method, a class of methods known as *conjugate gradient methods* have been devised (Polak, 1971, §2.3). The idea behind conjugate gradient methods is to choose the k^{th} search direction $\vec{s}_k$ in such a manner that $\vec{s}_k$ is conjugate to the $k-1$ previous search directions, implying that the minimization along $\vec{s}_k$ does not "spoil" the minimizations already performed along $\vec{s}_1, \vec{s}_2 \ldots, \vec{s}_{k-1}$. The reader is referred to Polak (1971, §2.3) for further details.

The minimization methods mentioned in this section are only effective if $Q(\vec{x})$ is unimodal. Methods applicable to a multimodal $Q(\vec{x})$ are described in §3.1.2.2.

3.1.2.2 Global Minimization Methods

Whereas numerous methods exist for finding the minimum of a unimodal function, there are very few methods available for finding the minimum of a multimodal function (Corana *et al.*, 1987). Such methods are termed *global minimization methods*. One class of global minimization methods, known as *multistart methods*, operate by performing a large number of minimum searches using some unimodal minimization technique, each search being started from a different point in the domain of interest, with the smallest minimum thus found being retained. However, such methods are ineffectual if the cost function possesses a large number of local minima in the domain of interest. Of all the global minimization methods currently available, a technique known as *simulated annealing* appears to be the most suitable method for problems in which one has little comprehension of the topology of the cost function involved.

Simulated annealing is now described in some detail, because it is later invoked in Chapter 6.

The simulated annealing algorithm is based on the method devised by Metropolis *et al.* (1953) for calculating averages of state functions in statistical mechanics. Specifically, the average $< f >$ of a function $f(\vec{s})$ of state $\vec{s}$, where $\vec{s}$ is a vector of large dimension which typically describes the coordinates of a number of atoms, is given by

$$< f > = \int f(\vec{s})\, e^{-E(\vec{s})/k_B T}\, d\vec{s} \;\Big/\; \int e^{-E(\vec{s})/k_B T}\, d\vec{s} \tag{3.11}$$

where $E(\vec{s})$ is the energy associated with state $\vec{s}$, and k_B is Boltzmann's constant. Metropolis *et al.* (1953) solve (3.11) by a Monte-Carlo technique (Hammersley and Handscomb, 1964, ch. 5,9), in which the sampling scheme is devised so that the majority of the sampling points correspond to small values of $E(\vec{s})$, since such points supply the major contribution to the integrals in (3.11). This sampling scheme turns out to be rather straightforward, as is now described.

At the k^{th} step, a small random displacement is made to the system, taking it from state $\vec{s}_k$ to state $\vec{s}'$ (this displacement typically involves the displacement of a single atom). The change in energy ΔE resulting from this displacement is then calculated, and the displacement is accepted (i.e. $\vec{s}_{k+1}$ is set to $\vec{s}'$) if $\Delta E \leq 0$. If $\Delta E > 0$, the displacement is accepted with probability $e^{-\Delta E/k_B T}$. If the displacement is not accepted, then $\vec{s}_{k+1}$ is simply set to $\vec{s}_k$. Finally, $f(\vec{s}_{k+1})$ is evaluated, and this value is incorporated into $< f >$. This procedure is repeated until the fluctuations in $< f >$ become sufficiently small.

The relevance of the Metropolis algorithm to global minimization has been realized by Pincus (1968), who shows that, for a continuous function $Q(\vec{x})$ with a unique global minimum $\vec{x}_{\min}$, the minimum is given by

$$\vec{x}_{\min} = \lim_{T \to 0} \int \vec{x}\, e^{-Q(\vec{x})/T}\, d\vec{x} \;\Big/\; \int e^{-Q(\vec{x})/T}\, d\vec{x} \tag{3.12}$$

Pincus (1970) points out that a global minimization problem can be solved by repeatedly applying the Metropolis algorithm with a succession of decreasing values of T. This method has been popularized by Kirkpatrick *et al.* (1983), who have coined the term *simulated annealing*. This name results from the analogy between this method and the process by which materials attain a low energy state through gradual cooling. Kirkpatrick *et al.* (1983) discuss the application of simulated annealing to combinatorial optimization problems (i.e. problems involving the minimization of functions of discrete variables).

The application of simulated annealing to the minimization of functions of continuous variables has been discussed by various authors (Bohachevsky *et al.*, 1986; Corana *et al.*, 1987). It is now appropriate to describe a general simulated annealing algorithm for the minimization of functions of continuous variables. The algorithm operates by generating a succession of points, the k^{th} of which is $\vec{x}_k$, where $1 \leq k \leq K$. The following steps are performed to generate the $(k+1)^{\text{th}}$ point:

Algorithm 3.1

Step 1: Randomly perturb $\vec{x}_k$ to $\vec{x}'$.

Step 2: Calculate $\Delta Q = Q(\vec{x}') - Q(\vec{x}_k)$.

Step 3: If $\Delta Q \leq 0$, accept the perturbation (i.e. set $\vec{x}_{k+1} = \vec{x}'$).

Step 4: If $\Delta Q > 0$, accept the perturbation with probability $e^{-\Delta Q/T}$.

T is now a control parameter, with the same units as $Q(\vec{x})$, but is still known as temperature due to the statistical-mechanical analogy. T is initially assigned a large value, and then steps 1-4 are repeated many times, during which the value of T is gradually decreased. When T is large, there is a high probability of accepting perturbations which increase $Q(\vec{x})$. The simulated annealing algorithm is thus unlikely to become trapped in local minima of $Q(\vec{x})$, in contrast to any unimodal minimization algorithm. As T is decreased, fewer perturbations which increase $Q(\vec{x})$ are accepted, until finally only perturbations which decrease $Q(\vec{x})$ are accepted. The final point $\vec{x}_K$ then lies near a minimum of $Q(\vec{x})$. Whether or not this is the global minimum of $Q(\vec{x})$ depends on many factors, such as the topology of $Q(\vec{x})$, the value of K, the method of generating the perturbations, and the manner in which T is reduced.

The manner in which T is varied as the algorithm is run is called the *annealing schedule* (Kirkpatrick *et al.*, 1983), which should be devised so that the "system" is in "equilibrium" at a particular value of T (i.e. $< Q(\vec{x}) >$ is effectively constant) before T is lowered. Although bounds on the annealing schedule have been proposed (Geman and Geman, 1984; Mitra *et al.*, 1986), the appropriate annealing schedule for a particular problem is usually determined by trial and error. Running the algorithm with $T = 0$ is known as quenching (due to its metallurgical analogy), which tends to be ineffectual for cost functions which possess numerous minima, since the algorithm usually becomes trapped in a minimum close to the starting point.

Besides the annealing schedule, the manner in which perturbations are generated is crucial to the effectiveness of the simulated annealing algorithm, when minimizing functions of continuous variables. The size of the perturbations (i.e. $\|\vec{x}' - \vec{x}_k\|$) is important, since few perturbations are accepted if the perturbations are large, thus wasting computational effort. Conversely, many perturbations are accepted if the perturbations are small, but this implies that the cost function surface is not being adequately explored. Metropolis *et al.* (1953) vary the size of the perturbations to ensure that approximately half of the perturbations are accepted on average. Various methods for adjusting the perturbation size are described by Corana *et al.* (1987). Perturbations are typically made along successive coordinate directions of $\vec{x}$, although they need not be. It is often advantageous to do so, however, since it may simplify the calculation of ΔQ. Note that the perturbations can be chosen so as to limit the minimization search to a restricted domain of $\vec{x}$ (e.g. to positive $\vec{x}$), thereby effecting a constrained minimization.

It is worth mentioning that various functions of $Q(\vec{x})$ and T may be devised which can replace $e^{-Q(\vec{x})/T}$ in (3.12). This suggests that similar algorithms to Algorithm 3.1 may be devised, which differ in their particular rules for accepting perturbations (known as *acceptance probability functions*). Anily and Federgruen (1987) state conditions which any global minimization algorithm of the simulated annealing type ought to satisfy, and they show how one can determine if an algorithm with a particular acceptance probability function complies with these conditions.

Areas in which the simulated annealing method has been applied are now listed, to indicate the power and generality of the method. Kirkpatrick *et al.* (1983) describe the application of simulated annealing to statistical mechanics (spin glasses in particular), combinatorial optimization (including the famous travelling salesman problem), and integrated circuit design. Sechen (1988) provides a thorough treatment of the application of simulated annealing to VLSI design. Brünger *et al.* (1987) have applied simulated annealing to crystallographic refinement. Image processing applications of simulated annealing include restoration of blurred and contaminated image (Geman and Geman, 1984), reconstruction from coded images (Smith *et al.*, 1983; Smith *et al.*, 1985), and phase retrieval (Nieto-Vesperinas and Mendez, 1986; Nieto-Vesperinas *et al.*, 1988). The application of simulated annealing to the phase problem is described further in §3.2.9.2, and a new application of simulated annealing, namely to the blind deconvolution problem, is presented in Chapter 6.

3.1.2.3 Regularization Methods

It was explained, in §3.1.2, how an image reconstruction problem could be posed as a minimization problem, the estimate $\hat{f}(\vec{x})$ of the original image $f(\vec{x})$ being that image which minimizes $Q\{\hat{f}(\vec{x})\}$ as defined by (3.10). Although this method of selecting $\hat{f}(\vec{x})$ has the advantage that it (usually) uniquely specifies $\hat{f}(\vec{x})$, the resulting $\hat{f}(\vec{x})$ often exhibits pronounced artefacts (such as ringing and high frequency noise) if the data $d(\vec{w})$ are appreciably contaminated.

An alternative approach to the image reconstruction problem is to accept as a *feasible image* any image which agrees (to within the estimated contamination level) with the data, rather than simply selecting that image which is in best agreement with the data, and then select $\hat{f}(\vec{x})$ from the set of feasible images according to some criterion. Such techniques are known as *regularization methods*. It is now convenient to re-express the equation for the measured data (3.9) in discrete form as

$$d[k] = T\{f[l]\}[k] + \sigma[k]\, n[k] \tag{3.13}$$

where $d[k]$ is the measured data (which comprises K samples), $T\{f[l]\}[k]$ describes the ideal data resulting from the original image $f[l]$ (which comprises L samples), $\sigma[k]$ is the standard deviation of the noise on $d[k]$, and $n[k]$ is a zero-mean random variable. Note that, although this discussion is restricted to one-dimensional data for notational convenience, the following results hold for data of any number of dimensions.

One common measure of the discrepancy between the measured data $d[k]$ and the ideal data $T\{\hat{f}[l]\}[k]$ resulting from $\hat{f}[l]$ (an estimate of $f[l]$) is the chi-squared statistic $\chi^2\{\hat{f}[l]\}$ given by (Burch *et al.*, 1983)

$$\chi^2\{\hat{f}[l]\} = \sum_{k=1}^{K} (T\{\hat{f}[l]\}[k] - d[k])^2/\sigma[k]^2 \tag{3.14}$$

To simplify the resulting constrained minimization problem, Burch *et al.* (1983) define the set of feasible images to be those $\hat{f}[l]$ for which

$$\chi^2\{\hat{f}[l]\} = K \tag{3.15}$$

although alternative definitions are, of course, possible.

To regularize an image reconstruction problem, one chooses the feasible image which optimizes some *regularization function*. One popular regularization function is known as *entropy*. The entropy of $\hat{f}[l]$ is denoted by $S\{\hat{f}[l]\}$, and is given by

$$S\{\hat{f}[l]\} = \sum_{l=1}^{L} s(\hat{f}[l]) \tag{3.16}$$

where the function $s(t)$ is usually either

$$s(t) = \log t \tag{3.17}$$

or

$$s(t) = -t \log t \tag{3.18}$$

Regularization methods in which $S\{\hat{f}[l]\}$ is maximized are called *maximum entropy methods*. It has been claimed by various authors (e.g. Ables, 1974; Shore and Johnson, 1980; Gull and Skilling, 1984), by statistical, thermodynamic and information theoretic arguments, that maximum entropy is in some sense the "optimum" regularization principle. It is often stated that the maximum entropy method produces solutions that are maximally non-committal with regard to absent data, or that there must be evidence in the data for any structure appearing in a maximum entropy reconstruction. Narayan and Nityananda (1986) provide a detailed critique of these claims.

Maximum entropy methods appear to perform well when reconstructing peaked images on flat backgrounds, but tend to perform poorly when reconstructing smooth, featureless images (Nityananda and Narayan, 1982). One explanation as to why maximum entropy methods can be successful is that both (3.17) and (3.18) have the following properties (Narayan and Nityananda, 1986):

$$\frac{ds}{dt} \to \infty \text{ as } t \to 0 \tag{3.19}$$

and

$$\frac{d^2 s}{dt^2} < 0 \tag{3.20}$$

The property (3.19) precludes pixels of $\hat{f}[l]$ from assuming negative values, and thus is equivalent to enforcing a positivity constraint upon $\hat{f}[l]$. The effect of (3.20) is to discourage the inclusion of ripples in the solution, and thus is equivalent to a smoothing constraint. Other functions which satisfy (3.19) and (3.20), such as $s(t) = \sqrt{t}$, have been shown to produce similar reconstructions to those obtained with (3.17) and (3.18) (Nityananda and Narayan, 1982).

The application of maximum entropy methods to the phase problem and blind deconvolution problem is discussed in §3.2.9.3 and §3.3.4 respectively.

3.2 Methods for Solving Phase Problems

Various methods for solving phase problems are described in this section. The first two subsections discuss methods for solving non-Fourier phase problems, namely the

crystallographic phase problem in (§3.2.1), and phase problems involving multiple intensity measurements (in §3.2.2). The remaining subsections are devoted to methods for solving Fourier phase problems. Methods which are only applicable for various special types of objects are described in §3.2.3, whilst methods for general objects are described in §§3.2.4–3.2.9. Phase retrieval methods which are based upon Hilbert transforms, the sampling theorem, polynomials and complex zeros are described in §§3.2.4, 3.2.5, 3.2.6 and 3.2.7 respectively. Iterative phase retrieval methods are detailed in §3.2.8, and finally the application of optimization methods to phase retrieval is described in §3.2.9.

3.2.1 Methods for the Crystallographic Phase Problem

The phase problem which arises in crystallography is described in §2.5.1, where the difference between the crystallographic and Fourier phase problem is explained. Although this thesis is only concerned in detail with Fourier phase problems, it is worth mentioning one class of methods, known as *direct methods*, for solving the crystallographic phase problem. These methods use solely the structure factor magnitude data and the *a priori* knowledge that the electron density is positive and localized, as opposed to other crystallographic methods which exploit various types of extra data. A concise history of direct methods is given by Woolfson (1987).

Direct methods operate in Fourier space, and so appropriate direct methods could well be incorporated into any Fourier phase retrieval technique which operates in Fourier space. By doing so, a positivity constraint, and (if appropriate) a localization constraint (which may be useful for astronomical images) could be implicitly enforced in image space.

3.2.2 Methods Requiring Multiple Intensity Measurements

Various phase retrieval techniques exist which rely on more than one intensity measurement. Two distinct classes of situation exist in which multiple intensity measurements are available. Phase retrieval methods for each class are now described.

Situations arise in which intensity can be measured in both image space and Fourier space (i.e. both $|f(\vec{x})|^2$ and $|F(\vec{u})|^2$ are available), as noted in §§2.5.2 and 2.5.4. The resulting phase problem can be solved by the algorithm of Gerchberg and Saxton (1972), which has already been mentioned (as an example of the iterative transform algorithm) in §3.1.1.1. In contrast to Fienup's error-reduction algorithm, the Gerchberg-Saxton algorithm only sometimes stagnates, and such stagnation can be easily overcome (Gerchberg, 1986).

Another class of situations involving multiple intensity measurements are those in which $|F(\vec{u})|^2$ and $|F'(\vec{u})|^2$ are measured, where $F'(\vec{u}) = F(\vec{u}) \odot D(\vec{u})$, with $D(\vec{u})$ being some known function. This occurs when measurements are made at different planes in an imaging system. It also arises in electron microscopy when images are recorded at differing degrees of defocus. The resulting phase problems can be solved by an iterative method due to Misell (1973). This method is similar to the Gerchberg-Saxton algorithm, with the forward and inverse Fourier transform steps being replaced by convolution with and deconvolution by $D(\vec{u})$ respectively. Recursive

methods for solving this type of phase problem have also been proposed (Dallas, 1976; Quatieri, 1981).

In a technique known as *exponential filtering*, $d(\vec{x})$ is chosen to be (for the one-dimensional case) $d(\vec{x}) = e^{kx}$, where k is a real constant. By comparing the positions of the complex zeros of $|F(\vec{u})|^2$ and $|F'(\vec{u})|^2$, the complex zeros of $F(u)$ can be straightforwardly separated from those of $F^*(u)$, thereby allowing the recovery of $f(x)$ (Walker, 1981). Although this technique can be extended to two dimensions, an iterative scheme similar to that of Misell's is typically applied to two-dimensional data (Walker, 1982).

3.2.3 Methods for Special Objects

This section describes a number of phase retrieval techniques which are applicable to objects of various special types. Methods for reconstructing objects which possess reference points are discussed in §3.2.3.1, whilst a technique for recovering objects which consist of a collection of points is detailed in §3.2.3.2.

3.2.3.1 Objects Possessing Reference Points

Consider an object $f(\vec{x})$ which may be usefully represented by

$$f(\vec{x}) = b(\vec{x}) + \delta(\vec{x} - \vec{a}) \tag{3.21}$$

where $\vec{a}$ is a constant position vector. The delta function in $f(\vec{x})$ is often termed a *reference point*. The significance of $f(\vec{x})$ containing a reference point is that the autocorrelation of $f(\vec{x})$ can then be expressed as

$$ff(\vec{x}) = bb(\vec{x}) + \delta(\vec{x}) + b(\vec{x} - \vec{a}) + b(\vec{a} - \vec{x}) \tag{3.22}$$

If $\vec{a}$ is such that $b(\vec{x}-\vec{a})$ does not overlap $bb(\vec{x})$ then $f(\vec{x})$ is said to satisfy a *holographic separation condition*, in which case it is evident from (3.22) that the phase problem is solved trivially (Bates and McDonnell, 1986, §21). In some practical situations it is possible to effectively add a reference point to an object, and in others it sometimes happens that a reference point exists (e.g. in astronomy, an appropriately located star).

Fienup (1983a) presents a method for reconstructing objects which possesses two or more of what he calls *latent reference points*, which are similar to holographic reference points, but need not be separated from the rest of the object by the holographic separation condition. This method initially computes object samples from the corresponding samples in the autocorrelation resulting from the latent reference points, then recursively computes the remaining object samples. However, this method has the disadvantage that the object support must belong to a special class. It should not be forgotten that recursive algorithms generally tend to be unstable in the presence of noise.

3.2.3.2 Objects Comprising Collections of Points

This section describes a method for directly reconstructing an image-form from a given autocorrelation, applicable in situations where the object comprises a collec-

tion of isolated point sources. Star clusters in optical astronomy are one example of such objects. The ingenious reconstruction method detailed below was devised by Fienup (Fienup *et al.*, 1982; Fienup, 1981a; Dainty and Fienup, 1987, §7.3A); a similar, though less sophisticated, scheme was reported by Christou (1981). Since this method forms the basis of the new blind deconvolution techniques introduced in Chapter 7, it is described here in some detail. This description follows that of Fienup (1982), who terms this technique *autocorrelation tri-intersection*, whereas Bates and McDonnell (1986, §21) refer to it as *triple autocorrelation holography*.

An object $f(\vec{x})$ comprising M isolated points, the m^{th} of which is located at $\vec{x}_m$ with amplitude f_m, is here represented by

$$f(\vec{x}) = \sum_{m=1}^{M} f_m \, \delta(\vec{x} - \vec{x}_m) \tag{3.23}$$

The support of $f(\vec{x})$ (see §1.5), here denoted by S_f, is then simply

$$S_f = \{\vec{x}_m : m = 1, \ldots, M\} \tag{3.24}$$

By substituting (3.23) into (1.29), the autocorrelation $a(\vec{x})$ of $f(\vec{x})$ is seen to be

$$a(\vec{x}) = \sum_{n=1}^{M} \sum_{m=1}^{M} f_n f_m \delta(\vec{x} - \vec{x}_m + \vec{x}_n) \tag{3.25}$$

Note that $a(\vec{x})$ possesses (at most) $M^2 - M + 1$ distinct points, and comprises M shifted and scaled versions (hereafter termed *translates*) of $f(\vec{x})$, or alternatively, comprises M translates of $f(-\vec{x})$. The technique described in this section involves combining translates of $a(\vec{x})$, in such a manner as to superimpose translates of $f(\vec{x})$ or $f(-\vec{x})$ contained therein.

The first step of the autocorrelation tri-intersection method consists of forming the *autocorrelation product* $a^{(2)}(\vec{x})$, given by

$$a^{(2)}(\vec{x}) = a(\vec{x}) \, a(\vec{x} - \vec{s}_1) \tag{3.26}$$

where $\vec{s}_1$ is the position of any point (where, from now on, *point* is taken to mean any part of the function in question which has non-zero value) in $a(\vec{x})$, apart from $\vec{0}$. From (3.25), it is evident that $\vec{s}_1$ must be of the form

$$\vec{s}_1 = \vec{x}_j - \vec{x}_k, \quad j \neq k, \quad \vec{x}_j, \vec{x}_k \in S_f \tag{3.27}$$

To simplify the analysis of the autocorrelation tri-intersection method, it is now assumed that the object comprises points of equal amplitudes, i.e.

$$f_m = 1, \quad m = 1, \ldots, M \tag{3.28}$$

Substitution of (3.27) and (3.25) into (3.26) yields

$$a^{(2)}(\vec{x}) = \sum_{n=1}^{M} \sum_{m=1}^{M} \delta(\vec{x} - \vec{x}_m + \vec{x}_n) \sum_{q=1}^{M} \sum_{p=1}^{M} \delta(\vec{x} - \vec{x}_p + \vec{x}_q - \vec{x}_j + \vec{x}_k) \tag{3.29}$$

Note that $a^{(2)}(\vec{x})$ only contains points at positions for which values of m, n, p, q, j and k exist that satisfy

$$\vec{x} = \vec{x}_m - \vec{x}_n = \vec{x}_p - \vec{x}_q + \vec{x}_j - \vec{x}_k \tag{3.30}$$

It is evident that (3.30) is satisfied when any of the following relationships arise:

$$m = p \quad \text{and} \quad n = k \quad \text{and} \quad q = j \tag{3.31a}$$
$$m = j \quad \text{and} \quad n = q \quad \text{and} \quad p = k \tag{3.31b}$$
$$m = n \quad \text{and} \quad p = k \quad \text{and} \quad q = j \tag{3.31c}$$
$$m = j \quad \text{and} \quad n = k \quad \text{and} \quad p = q \tag{3.31d}$$

Thus (3.29) can be expressed as

$$\begin{aligned} a^{(2)}(\vec{x}) &= \sum_{m=1}^{M} \delta(\vec{x} - \vec{x}_m + \vec{x}_k) + \sum_{n=1}^{M} \delta(\vec{x} + \vec{x}_n - \vec{x}_j) \\ &+ (M-1)\,\delta(\vec{x}) + (M-1)\,\delta(\vec{x} - \vec{x}_j + \vec{x}_k) \\ &+ \text{other terms} \end{aligned} \tag{3.32}$$

where the first four terms of (3.32) result from (3.31a,b,c,d) respectively, and the other terms arise if any further combinations of m, n, p, q, j and k exist which satisfy (3.30). It is necessary to precisely state the conditions under which no other terms appear in (3.32), as follows:

$$\begin{aligned} &\text{whenever} && \vec{x}_a, \vec{x}_b, \vec{x}_c, \vec{x}_d, \vec{x}_e, \vec{x}_f \in S_f \\ &\text{and} && \vec{x}_a - \vec{x}_b + \vec{x}_c - \vec{x}_d + \vec{x}_e - \vec{x}_f = \vec{0} \text{ and } a \neq b \\ &\text{then} && a = d \text{ or } a = f \text{ and } b = c \text{ or } b = e \end{aligned} \tag{3.33}$$

Condition (3.33), termed a *nonredundancy condition*, is violated if the points contained in any subset of S_f happen to be juxtaposed in any of the following ways:

1. Two pairs of points possess the same vector separation.
2. The vector separation of one pair of points is twice that of another pair.
3. The vector separations of three pairs of points sum to zero.

Inspection of (3.32) reveals that $a^{(2)}(\vec{x})$ contains a translate of $f(\vec{x})$ and a translate of $f(-\vec{x})$. This is intuitively reasonable, since shifting $a(\vec{x})$ by an amount $\vec{s}_1$ serves to superimpose the two translates of $f(\vec{x})$ contained in $a(\vec{x})$ which are separated by $\vec{s}_1$. However, since $a(\vec{x})$ also contains two translates of $f(-\vec{x})$ which are separated by $\vec{s}_1$, these are also superimposed. The next step of the autocorrelation tri-intersection method consists of forming the *autocorrelation triple product* $a^{(3)}(\vec{x})$, given by

$$a^{(3)}(\vec{x}) = a^{(2)}(\vec{x})\, a(\vec{x} - \vec{s}_2) \tag{3.34}$$

where $\vec{s}_2$ is the position of any point in $a^{(2)}(\vec{x})$, apart from $\vec{0}$ or $\vec{s}_1$. From (3.32), it is evident that $\vec{s}_1$ must (provided condition (3.33) is satisfied) either be of the form

$$\vec{s}_2 = \vec{x}_l - \vec{x}_k, \quad l \neq k, \quad \vec{x}_l \in S_f \tag{3.35}$$

or

$$\vec{s}_2 = \vec{x}_j - \vec{x}_l, \quad l \neq j, \quad \vec{x}_l \in S_f \tag{3.36}$$

Note that $\vec{x}_j$ and $\vec{x}_k$ in (3.36) and (3.35) are defined by (3.27). Substitution of (3.35) and (3.32) and (3.25) into (3.34) yields (after some manipulation, of the sort involved in obtaining (3.32) from (3.29))

$$\begin{aligned} a^{(3)}(\vec{x}) &= \sum_{m=1}^{M} \delta(\vec{x} - \vec{x}_m + \vec{x}_k) \\ &+ (N-1)[\delta(\vec{x}) + \delta(\vec{x} - \vec{x}_j + \vec{x}_k) + \delta(\vec{x} - \vec{x}_l + \vec{x}_k)] \\ &+ \text{other terms} \end{aligned} \tag{3.37}$$

where, once again, the other terms arise only if condition (3.33) is violated. If $\vec{s}_2$ is of the form (3.36) then $a^{(3)}(\vec{x})$ is given by

$$\begin{aligned} a^{(3)}(\vec{x}) &= \sum_{m=1}^{M} \delta(\vec{x} + \vec{x}_m - \vec{x}_j) \\ &+ (N-1)[\delta(\vec{x}) + \delta(\vec{x} - \vec{x}_j + \vec{x}_k) + \delta(\vec{x} - \vec{x}_j + \vec{x}_l)] \\ &+ \text{other terms} \end{aligned} \tag{3.38}$$

The significance of (3.37) and (3.38) is that $a^{(3)}(\vec{x})$ consists solely (provided condition (3.33) is satisfied) of a translate of either $f(\vec{x})$ or $f(-\vec{x})$ (note that the second terms of both RHS (3.37) and RHS (3.38) represent points at $\vec{0}$, $\vec{s}_1$ and $\vec{s}_2$, which all coincide with points of the translate of $f(\vec{x})$ or $f(-\vec{x})$). It can be made clear why this should be so by expressing $a^{(3)}(\vec{x})$ as

$$a^{(3)}(\vec{x}) = [a(\vec{x})\, a(\vec{x} - \vec{s}_1)] \; [a(\vec{x})\, a(\vec{x} - \vec{s}_2)] \tag{3.39}$$

For the reasons stated in the discussion preceding (3.32), it is evident that both of the autocorrelation products in RHS (3.39) contain a translate of $f(\vec{x})$ and a translate of $f(-\vec{x})$. Selecting $\vec{s}_2$ in the manner described in the sentence containing (3.34) ensures that either the two translates of $f(\vec{x})$ or the two translates of $f(-\vec{x})$ are superimposed, depending on whether $\vec{s}_2$ happens to satisfy (3.35) or (3.36).

The above description of the autocorrelation tri-intersection method has been restricted (for expositional convenience) to objects comprising points of equal amplitudes. However, it is straightforward (Fienup, 1982) to show that, for an object comprising points of differing amplitudes, $a^{(3)}(\vec{x})$ contains a translate of either $[f(\vec{x})]^3$ or $[f(-\vec{x})]^3$. By taking the cube root of the amplitudes of the points in $a^{(3)}(\vec{x})$, the amplitudes of the points of an image-form of $f(\vec{x})$ can be recovered (apart from those at $\vec{0}$, $\vec{s}_1$ and $\vec{s}_2$, but these can be obtained by a few further simple manipulations).

It is worthwhile considering further the situations in which the other terms appear in (3.37) or (3.38) (i.e. situations when $a^{(3)}(\vec{x})$ includes points other than those contained in a translate of $f(\vec{x})$ or $f(-\vec{x})$, here termed *spurious points*). It is guaranteed that $a^{(3)}(\vec{x})$ contains no spurious points if $f(\vec{x})$ satisfies condition (3.33). However, $f(\vec{x})$ can violate condition (3.33) and $a^{(3)}(\vec{x})$ still remain free from spurious points, if $\vec{s}_1$ and $\vec{s}_2$ happen to be chosen so as not to include the points which violate (3.33). Furthermore, spurious points in $a^{(3)}(\vec{x})$ often appear in different locations for

different choices of $\vec{s}_1$ and $\vec{s}_2$. In such cases, the spurious points can be removed by appropriately combining several autocorrelation triple products, each formed with different choices of $\vec{s}_1$ and $\vec{s}_2$ (Fienup, 1982, see also Crimmins *et al.*, 1990).

In practice, a measured autocorrelation must be diffraction-limited, implying that the autocorrelation points are of finite size. Thus the spacings between pairs of autocorrelation points must be appreciably greater than the size of each autocorrelation point if autocorrelation tri-intersection is to be successful.

3.2.4 Hilbert Transform Methods

The use of logarithmic Hilbert transforms to effect one-dimensional phase retrieval has been examined by various authors (e.g. Burge *et al.*, 1976; Ross *et al.*, 1978). However, while Nieto-Vesperinas (1980) shows that the form of the two-dimensional logarithmic Hilbert transform is unsuitable for computing phase from magnitude, he points out that one-dimensional logarithmic Hilbert transforms can be applied to strips of a two-dimensional Fourier intensity. This is the approach taken by Nakajima and Asakura (1986), but their technique requires extra information, in the form of intensity measurements obtained with an exponential filter.

3.2.5 Sampling Theorem Based Methods

Two phase retrieval techniques, which are both based upon the sampling theorem (§1.5), are now described. A method which attempts to find a solution exactly satisfying the sampling theorem in Fourier space is presented in §3.2.5.1, whereas §3.2.5.2 describes a method which only attempts to satisfy the sampling theorem approximately.

3.2.5.1 Exact Sampling Theorem Method

The phase retrieval method which is about to be described was first reported by Arsenault and Chalasinska-Macukow (1983). An alternative implementation of this method has been presented by Chalasinska-Macukow and Arsenault (1985), which reduces the method's computational requirements.

Consider a two-dimensional compact image $f(x,y)$, with extent L in both the x and y directions. It follows from the sampling theorem (see §1.5) that $F(u,v)$ may be represented by

$$F(u,v) = \sum_{p=-N}^{N} \sum_{q=-N}^{N} F_{p,q} \operatorname{sinc}(Lu - p) \operatorname{sinc}(Lv - q) \tag{3.40}$$

Recall that the $F_{p,q}$ (of which there are $(2N+1)^2$) are the *Nyquist samples* of $F(u,v)$ (existing at the *Nyquist sample points*, which are spaced by the *Nyquist spacing* in the u and v directions) given by

$$F_{p,q} = F\left(\frac{p}{L}, \frac{q}{L}\right) \tag{3.41a}$$

$$= M_{p,q}\, e^{i\,\Phi_{p,q}} \tag{3.41b}$$

where $M_{p,q}$ and $\Phi_{p,q}$ are introduced for later convenience. It is assumed that samples of $|F(u,v)|$ are available on a grid of points spaced by half the Nyquist spacing (i.e. spaced by $1/2L$) in both the u and v directions, for the reasons stated in §2.1.

Some useful terms, as invoked by Bates (1982b, see also Bates and McDonnell, 1986, §22), are now defined. The sample points located between the Nyquist sample points are here called *in-between sample points*. The samples of $F(u,v)$ located midway between $F_{p,q}$ and $F_{p+1,q}$, $F_{p,q+1}$ and $F_{p+1,q+1}$ are conveniently named the *in-row, in-column*, and *diagonal* in-between samples respectively, and are denoted by $F_{p+1/2,q}$, $F_{p,q+1/2}$ and $F_{p+1/2,q+1/2}$ respectively. The available data for the Fourier phase problem is thus four sets of magnitude samples (with each set comprising $(2N+1)^2$ members), namely the Nyquist, in-row in-between, in-column in-between, and diagonal in-between magnitude samples.

Substituting (3.41b) into (3.40) and multiplying (3.40) by its complex conjugate yields

$$\begin{aligned} |F(u,v)|^2 &= \left[\sum_{p=-N}^{N}\sum_{q=-N}^{N} M_{p,q}\,\mathrm{sinc}(Lu-p)\,\mathrm{sinc}(Lv-q)\,\cos\Phi_{p,q}\right]^2 \\ &+ \left[\sum_{p=-N}^{N}\sum_{q=-N}^{N} M_{p,q}\,\mathrm{sinc}(Lu-p)\,\mathrm{sinc}(Lv-q)\,\sin\Phi_{p,q}\right]^2 \end{aligned} \tag{3.42}$$

By inserting the values of $(2N+1)^2$ in-between magnitude samples for $|F(u,v)|$ in (3.42), $(2N+1)^2$ nonlinear equations in the $(2N+1)^2$ variables $\Phi_{p,q}$ may be obtained. Arsenault and Chalasinska-Macukow (1983) solve this set of equations by a Newton-Raphson method, but the solution thus obtained is unfortunately not unique (i.e. it depends upon the starting point used by the equation-solving routine).

The non-uniqueness of solution of this method may appear to be at odds with the uniqueness arguments for two-dimensional Fourier phase problems presented in §2.4.2. This is not so, however, since only half of the available $4(2N+1)^2$ magnitude samples are employed to find the solution. The values of $|F(u,v)|$ (evaluated by substituting the calculated values of $\Phi_{p,q}$ into (3.42)) at the in-between sample points other than those used to calculate the solution are likely to differ significantly from the given in-between magnitude samples.

The reason for the non-uniqueness of the above method is clarified by considering the effect of using the set of Nyquist magnitude samples in conjunction with, say, the set of in-row in-between magnitude samples to solve (3.42). The in-row in-between magnitude samples only exist at (u,v) coordinates for which $v = q/L$, where q is an integer. In this case, (3.42) reduces to

$$\begin{aligned} |F(u,q/L)|^2 &= \left[\sum_{p=-N}^{N} M_{p,q}\,\mathrm{sinc}(Lu-p)\,\cos\Phi_{p,q}\right]^2 \\ &+ \left[\sum_{p=-N}^{N} M_{p,q}\,\mathrm{sinc}(Lu-m)\,\sin\Phi_{p,q}\right]^2 \end{aligned} \tag{3.43}$$

The two-dimensional phase problem is thereby reduced to $2N+1$ one-dimensional phase problems, each with $2N+1$ unknown phases. This is precisely the means

employed by Chalasinska-Macukow and Arsenault (1985) to reduce the computational requirements of this method. However, each of these one-dimensional phase problems possesses 2^{2N} solutions, for the reasons stated in §2.4.1.

To avoid multiple solutions, it appears that more than $2(2N+1)^2$ of the total $4(2N+1)^2$ available magnitude samples must be employed. However, doing so results in a set of overdetermined non-linear equations in the $2N+1$ variables $\Phi_{p,q}$. Such a set of equations is typically solved by methods which minimize a cost function comprising the sum of squared residuals (Dennis and Scnabel, 1983, ch. 10). If this cost function possesses local minima (which is likely) then such methods tend to find poor solutions to the set of equations, since they become trapped in these local minima.

3.2.5.2 Crude Phase Estimation

This section describes a phase retrieval technique, commonly referred to as *crude phase estimation*, which is related to, but has distinct advantages over, that of Arsenault and Chalasinska-Macukow (1983). This technique was first reported by Bates (1982b) and demonstrated by Fright and Bates (1982), but is also described by Fright (1984, ch. 5) and Bates and Mnyama (1986, §V.B). The notation developed in §3.2.5.1 is now invoked to describe this method.

To overcome the problems experienced with the systems of non-linear equations in the method described in §3.2.5.1, the sinc function $\mathrm{sinc}(u)$ in (3.40) is approximated by an appropriate sampling function $\mathrm{samp}(u)$. Since it is assumed that the samples of $|F(u,v)|$ are spaced by $1/2L$, it is only necessary to define $\mathrm{samp}(u)$ at the points $u = k/2$, where k is an integer. A useful choice of $\mathrm{samp}(k/2)$ is

$$\mathrm{samp}\left(\frac{k}{2}\right) = \begin{cases} 1 & \text{if } k = 0 \\ \beta & \text{if } k = \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.44}$$

where β is a constant, typically ≈ 0.5. On replacing $\mathrm{sinc}(u)$ in (3.40) with $\mathrm{samp}(u)$ and substituting $u = (p+\frac{1}{2})/L$ and $v = q/L$, it follows that

$$F_{p+\frac{1}{2},q} \approx \beta\,(F_{p,q} + F_{p+1,q}) \tag{3.45}$$

and similarly

$$F_{p,q+\frac{1}{2}} \approx \beta\,(F_{p,q} + F_{p,q+1}) \tag{3.46}$$

Now since the magnitudes of each of the three samples appearing in (3.45) are known, $|\Phi_{p+1,q} - \Phi_{p,q}|$ can be straightforwardly estimated from (3.45). In a similar manner, $|\Phi_{p,q+1} - \Phi_{p,q}|$ can be estimated from (3.46). It is now convenient to introduce the notation $\Psi^{a,b}_{c,d}$ for the estimate of $|\Phi_{a,b} - \Phi_{c,d}|$ obtained from (3.45) or (3.46), and $\hat{\Phi}_{a,b}$ for an estimate of $\Phi_{a,b}$.

Imagine now that $\Psi^{1,0}_{0,0}$, $\Psi^{0,1}_{0,0}$, $\Psi^{1,1}_{0,1}$ and $\Psi^{1,1}_{1,0}$ have been determined from (3.45) and (3.46). Because it is only possible to recover an image-form of $f(x,y)$, one can arbitrarily set $\hat{\Phi}_{0,0} = 0$ and $\hat{\Phi}_{1,0} = \Psi^{1,0}_{0,0}$. Only two possible choices of $\hat{\Phi}_{0,1}$ and $\hat{\Phi}_{1,1}$ then remain, namely $\hat{\Phi}_{0,1} = \pm\Psi^{0,1}_{0,0}$ and $\hat{\Phi}_{1,1} = \hat{\Phi}_{1,0} \pm \Psi^{1,1}_{1,0}$. If equality held in (3.45) and (3.46), then only one (in general) of the four possible combinations of $\hat{\Phi}_{0,1}$ and $\hat{\Phi}_{1,1}$ would satisfy $|\hat{\Phi}_{1,1} - \hat{\Phi}_{0,1}| = \Psi^{1,1}_{0,1}$. In practice, one can simply choose the $\hat{\Phi}_{0,1}$ and

$\hat{\Phi}_{1,1}$ for which $|\hat{\Phi}_{1,1} - \hat{\Phi}_{0,1}|$ is closest to $\Psi_{0,0}^{1,0}$. Then $\hat{\Phi}_{0,2}$ and $\hat{\Phi}_{1,2}$ can be determined in a similar manner, the process being repeated until all $2N+1$ of the $\hat{\Phi}_{p,q}$ have been determined.

Fright and Bates (1982) have shown that this method produces useful image estimates for small, simple images, and is robust in the presence of data contamination. However, its performance degrades with increased image size and complexity, because samp(u) is only a crude approximation to sinc(u). It has been demonstrated (Bates and Fright, 1983) that the convergence of Fienup's algorithms is significantly accelerated by starting with an image produced by crude phase estimation.

It is interesting to note that the crude phase estimation algorithm, unlike that of Arsenault and Chalasinska-Macukow (1983), does not appear to be troubled by non-uniqueness of solution. This observation can be explained by the fact that the former algorithm uses $\frac{3}{4}$ (as opposed to $\frac{1}{2}$ for the latter) of the available $(2M+1)^2$ magnitude samples.

3.2.6 Polynomial Based Methods

It is convenient to consider a two-dimensional discrete image $f[m,n]$ which comprises $M \times M$ pixels, the autocorrelation of which is $a[m,n]$. Recall from §2.4.2 that the z-transform of $a[m,n]$, denoted by $A(\zeta,\gamma)$ (which is a polynomial in ζ and γ), is the product of $F(\zeta,\gamma)$ (the z-transform of $f[m,n]$) and $\mathcal{F}^*(1/\zeta^*, 1/\gamma^*)$. The problem of reconstructing $f[m,n]$ from $a[m,n]$ is thus equivalent to that of factorizing $A(\zeta,\gamma)$. Various authors (Berenyi *et al.*, 1985; Nieto-Vesperinas and Dainty, 1986; Shi and Bose, 1988) have considered using bivariate polynomial factorization methods to factorize $A(\zeta,\gamma)$.

However, the practical problem faced by any technique which relies on polynomial factorization is that contamination of $a[m,n]$ causes $A(\zeta,\gamma)$ to become irreducible (i.e. unfactorizable) (Sanz and Huang, 1985). Evidently, some method of finding an approximate factorization of an irreducible polynomial would be desirable. Note that such approximate factorizations do exist, since they are effectively what are computed by iterative phase retrieval algorithms. Alternatively, some method of finding a reducible polynomial which is in some sense close to a given irreducible polynomial would be valuable.

3.2.7 Complex Zero Based Methods

This section describes various phase retrieval techniques which involve the manipulation of the complex zeros (see §§2.4.1 and 2.4.2) of an image's spectrum. Methods for phase recovery which rely on the tracking of complex zeros are outlined in §3.2.7.1, whereas §3.2.7.2 describes a method which involves complex zero flipping.

3.2.7.1 Zero-Tracking Methods

The first description of a phase retrieval method involving the tracking of complex zeros was presented by Napier and Bates (1974). This technique involves computing the zeros of a radial slice through the Fourier intensity and then tracking these zeros

as the slice angle is varied. However, this technique is only applicable to positive images.

Lane *et al.* (1987) present a phase retrieval algorithm that is based upon the concept of the zero-sheet (introduced in §2.4.2 as an argument for the uniqueness of multi-dimensional phase problems), which is now described. Izraelevitz and Lim (1987) report an algorithm which tracks zeros in a similar manner, but then uses a different method to recover the image.

Once again, it is convenient to consider the discrete $M \times M$ pixel image $f[m,n]$ with autocorrelation $a[m,n]$, with respective z-transforms $\mathcal{F}(\zeta,\gamma)$ and $\mathcal{A}(\zeta,\gamma)$. By fixing the value of ζ at some (complex) constant value ζ', $\mathcal{A}(\zeta,\gamma)$ can be reduced to a polynomial in γ alone, which is here written as $\mathcal{A}'(\gamma)$. Similarly, $\mathcal{A}(\zeta,\gamma)$ is reduced to $\mathcal{A}''(\zeta)$ by setting $\gamma = \gamma'$. It follows from §2.4.1 that $\mathcal{A}'(\gamma)$ possesses $2M-2$ complex zeros, which are arranged in $M-1$ conjugate reciprocal pairs. Now because of the analyticity of $\mathcal{A}(\zeta,\gamma)$, the positions of the zeros of $\mathcal{A}'(\gamma)$ migrate continuously through γ-space if ζ' is varied in a continuous manner. The zeros of $\mathcal{A}'(\gamma)$ can thus be tracked as ζ' is varied, thereby mapping out the zero-sheet of $\mathcal{A}(\zeta,\gamma)$.

By varying ζ' around appropriate closed paths in ζ-space, it is possible to arrange the zeros of $\mathcal{A}'(\gamma)$ into various groups, where the members of each group lie on the same zero-sheet. If there are only two such groups (which is almost always the case), then $f[m,n]$ and $f[-m,-n]$ can be straightforwardly recovered. Lane *et al.* (1987) demonstrate the effectiveness of their algorithm by recovering a 16×16 pixel complex image from its autocorrelation.

It was mentioned in §2.4.3 that the addition of contamination to $a[m,n]$ results in $\mathcal{A}(\zeta,\gamma)$ becoming irreducible. The zero-sheet of $\mathcal{A}(\zeta,\gamma)$ is then a single sheet, comprising the distorted zero-sheets of $\mathcal{F}(\zeta,\gamma)$ and $\mathcal{F}^*(1/\zeta^*,1/\gamma^*)$, which become joined at points called *bridges*. The number of bridges tends to increase with increased contamination (Lane and Bates, 1987a). It is not obvious at present how to effect phase retrieval via zero-sheets when the data are contaminated. However, the insight offered by Bates *et al.* (1990) into the effect of contamination on zero-sheets may eventually lead to useful phase retrieval algorithms based on zero-sheets.

3.2.7.2 Zero-Flipping Methods

The possibility of solving the two-dimensional phase problem by generating all possible solutions via zero-flipping was suggested by Bates (1982b). Deighton *et al.* (1985) have demonstrated the reconstruction of a 16×16 pixel image by such a technique, but provide scant details of their algorithm. The method is now outlined, using the notation developed in §3.2.7.1.

It is convenient to consider values of ζ' and γ' (the values at which ζ and γ are fixed) satisfying

$$\zeta' = e^{i2\pi p/M} \tag{3.47a}$$

$$\gamma' = e^{i2\pi q/M} \tag{3.47b}$$

where p and q are integers. The reason for choosing such values is made clear by

substituting (3.47) into the expression (1.69) for $\mathcal{F}(\zeta, \gamma)$, which yields

$$\begin{aligned} \mathcal{F}(\zeta', \zeta') &= \sum_m \sum_n f[m,n]\, e^{i2\pi(mp+nq)/M} \\ &= F[p,q] \end{aligned} \tag{3.48}$$

where $F[p,q]$ is the DFT (see §1.6) of $f[m,n]$. Setting $p = 0$ fixes ζ', thereby allowing the zeros of $\mathcal{A}'(\gamma)$ to be computed. By zero-flipping (§2.4.1), the 2^{M-1} polynomials $\mathcal{P}(\gamma)$ which satisfy $\mathcal{P}(\gamma)\mathcal{P}^*(1/\gamma^*) = \mathcal{A}'(\gamma)$ can be inferred.

Evaluating a particular $\mathcal{P}(\gamma)$ at the points $\gamma = \gamma'$ as given by (3.47b) for $q = 0, 1, \ldots, M-1$, provides a possible one-dimensional solution for $F[0,q]$. By repeating this procedure for all 2^{M-1} of the $\mathcal{P}(\gamma)$, 2^{M-1} possible $F[0,q]$ can be determined. In a similar manner, 2^{M-1} possibilities for each of $F[1,q]$, $F[p,0]$ and $F[p,1]$ can be computed. But because the summed phase differences around the loop comprising $F[0,0]$, $F[1,0]$, $F[1,1]$ and $F[0,1]$ must be a multiple of 2π (a requirement known as *phase closure*), there are usually only two possible combinations (corresponding to $f[m,n]$ and $f[-m,-n]$) of $F[0,q]$, $F[1,q]$, $F[p,0]$ and $F[p,1]$ which are mutually consistent. Only those which are consistent need be retained, and the remaining phases of $F[p,q]$ can then be similarly determined.

The computational demands of this phase retrieval technique are large, since $2^{4(M-1)}$ comparisons must be made to determine the first four phases. Furthermore, exact phase closure never occurs in practice, because of the inevitable contamination of the data. Lane (1988, §7.2) suggests that this technique may be made less computationally demanding, and robust in the presence of noise, by only using a small number of low-frequency zeros and accepting approximate phase closure.

Canterakis (1983) has also proposed a phase retrieval technique which employs zero-flipping. However, this method is not discussed further here, since the reconstruction of an $M \times M$ image requires calculation of 2^{2M^2} candidate images, which is computationally impractical.

3.2.8 Iterative Methods

This section describes various iterative methods for solving the Fourier phase problem. A class of methods devised by J. R. Fienup, which are hereafter referred to collectively as Fienup's algorithms, are first described. Fienup's error-reduction algorithm is introduced in §3.2.8.1, whilst §3.2.8.2 describes the more powerful input-output algorithms. Important details regarding the practical application of Fienup's algorithms are given in §3.2.8.3. Finally, the application to the phase problem of a generalization of the method of projections onto convex sets, termed generalized projections, is described in §3.2.8.4.

3.2.8.1 The Error Reduction Algorithm

The simplest of Fienup's (1982) phase retrieval algorithms, which is known as the *error-reduction* or *error correction algorithm* (for reasons which are soon explained), is simply a special case of the general iterative transform algorithm described in §3.1.1.1. Specifically, the error-reduction algorithm results from the incorporation of

the following constraints (as listed in Table 3.1) into the iterative transform algorithm of Figure 3.1 : image positivity, image support and given Fourier magnitude. The rationale behind the error-reduction algorithm is identical to that given in §3.1.1.1 for the general iterative transform algorithm, and so is not repeated here.

It is convenient to describe the error-reduction algorithm in terms of the same variables appearing in Figure 3.1 and Table 3.1. However, a discrete two-dimensional image notation (in which it is assumed that all images possess $M \times M$ pixels) is employed, to emphasize the computational aspects of the algorithm. The error-reduction algorithm is then defined by the following four steps, where variables subscripted with a k denote quantities at the k^{th} iteration:

$$\tilde{F}_k[p,q] = \text{FFT}\{\tilde{f}_k[m,n]\} \tag{3.49a}$$

$$\bar{F}_k[p,q] = M[p,q]\,\exp(i\,\text{Ph}\{\tilde{F}_k[p,q]\}) \tag{3.49b}$$

$$\bar{f}_k[m,n] = \text{IFFT}\{\bar{F}_k[p,q]\} \tag{3.49c}$$

$$\tilde{f}_{k+1}[m,n] = \begin{cases} \bar{f}_k[m,n] & \text{if } [m,n] \notin V_k \\ 0 & \text{if } [m,n] \in V_k \end{cases} \tag{3.49d}$$

$M[p,q]$ here denotes the measured magnitude data, and V_k denotes the set of pixels at which $\bar{f}_k[m,n]$ violates the image space constraints. Thus, if a positivity constraint is employed, V_k is the set of pixels at which $\bar{f}_k[m,n] < 0$. If a support constraint is employed, and implemented with an array called the *image mask*, denoted by $mask[m,n]$, which is unity at pixels within the estimated image support and zero elsewhere, then V_k is the set of pixels at which $mask[m,n] = 0$.

Useful measures of the convergence of the algorithm are provided by the *normalized squared image space error* (usually referred to simply as the *image error*) at the k^{th} iteration, denoted by E_k^{I}, given by

$$E_k^{\text{I}} = \sum_{[m,n]\in V_k}\sum |\bar{f}_k[m,n]|^2 \Big/ \sum_m \sum_n |\bar{f}_k[m,n]|^2 \tag{3.50}$$

and the *normalized squared Fourier space error* (or simply *Fourier error*) E_k^{F}, given by

$$E_k^{\text{F}} = \sum_p \sum_q \Big||\tilde{F}_k[p,q]| - M[p,q]\Big|^2 \Big/ \sum_p \sum_q M[p,q]^2 \tag{3.51}$$

Note that all summations are from 1 to M unless otherwise specified. It is straightforward to show (Fienup, 1982) that

$$E_{k+1}^{\text{F}} \leq E_k^{\text{I}} \leq E_k^{\text{F}} \tag{3.52}$$

The significance of (3.52) is that the Fourier (or image) error can never increase from iteration to iteration, which is why the error-reduction algorithm is so called.

Fienup (1982) has shown that the error-reduction algorithm is closely related to the steepest descent method (see §3.1.2.1). This is demonstrated by considering the phase problem in terms of how to find the values of the M^2 pixels of $\hat{f}[m,n]$ (an estimate of $f[m,n]$) which minimize the cost function $Q\{\hat{f}[m,n]\}$, as given by

$$Q\{\hat{f}[m,n]\} = \sum_p \sum_q \Big||\hat{F}[p,q]| - M[p,q]\Big|^2 \tag{3.53}$$

It transpires (Fienup, 1982) that the gradient of $Q\{\hat{f}[m,n]\}$ at the point $\tilde{f}_k[m,n]$, which is here denoted by $\nabla Q_k[m,n]$, is simply given by

$$\begin{aligned}\nabla Q_k[m,n] &= \left.\frac{\partial Q\left\{\hat{f}[m,n]\right\}}{\partial \hat{f}[m,n]}\right|_{\hat{f}[m,n]=\tilde{f}_k[m,n]} \\ &= 2\left[\tilde{f}_k[m,n]-\bar{f}_k[m,n]\right] \end{aligned} \tag{3.54}$$

Performing the first three steps (3.49a–c) of the error-reduction algorithm is thus equivalent to minimizing $Q\{\hat{f}[m,n]\}$ by taking a steepest descent step from the point $\tilde{f}_k[m,n]$. Hence one iteration of the error-reduction algorithm may be considered as a steepest descent minimization of $Q\{\hat{f}[m,n]\}$ (which in fact reduces $Q\{\hat{f}[m,n]\}$ to zero) followed by a step in which $\bar{f}[m,n]$ is forced to meet the image space constraints.

Despite the relationship (3.52), the decrease in E_k^{F} or E_k^{I} usually becomes extremely slow after a number of iterations of the error-reduction algorithm have been performed, rendering the algorithm ineffective in reconstructing all but the simplest of images. When the decrease in E_k^{F} or E_k^{I} becomes insignificant, the algorithm is said to have *stagnated*. An example presented by Fienup (1982) indicates that the error-reduction algorithm can sometimes work its way out of stagnation (but usually only after an extremely large number of iterations), allowing the error to fall significantly before again stagnating. This is an important point to note, because it suggests that modifications to the error-reduction algorithm (e.g. the incorporation of over-relaxation) may significantly accelerate its convergence.

A phase retrieval algorithm reported by Seldin and Fienup (1989), which is derived from the iterative blind deconvolution algorithm (see §3.3.3.2), bears a striking similarity to the error-reduction algorithm, and so is now described. It is not at all surprising that a blind deconvolution algorithm should find application in phase retrieval, since the phase problem is merely a special case of the blind deconvolution problem, as explained in §2.6. Seldin and Fienup (1989) only use one half of the iterative blind deconvolution algorithm loop. This is because the standard blind deconvolution algorithm, when used for phase retrieval, provides two estimates of the desired image per iteration (i.e. provides an estimate of $f(\vec{x})$ and of $f^*(-\vec{x})$.

It transpires that Seldin and Fienup's (1989) algorithm is equivalent to the error-reduction algorithm (3.49), with (3.49b) replaced by

$$\bar{F}_k[p,q] = \tilde{F}_k[p,q]\,\frac{M[p,q]^2}{|\tilde{F}_k[p,q]|^2+\Phi} \tag{3.55}$$

where Φ is some constant. The comparison of (3.55) with (3.49b) is aided by expressing (3.49b) as

$$\bar{F}_k[p,q] = \tilde{F}_k[p,q]\,\frac{M[p,q]}{|\tilde{F}_k[p,q]|} \tag{3.56}$$

It is then evident that Seldin and Fienup's (1989) algorithm only differs from the error-reduction algorithm in that the Fourier domain constraint is over-relaxed, and incorporates a Wiener filter. Consequently, this algorithm performs similarly to the error-reduction algorithm, but tends to produce smoother reconstructions at higher noise levels, due to the Wiener filter in (3.55) (Seldin and Fienup, 1989).

3.2.8.2 Input-Output Algorithms

This section describes a class of algorithms which have been devised to overcome the stagnation problems experienced by the error-reduction algorithm. Consider the change in $\bar{f}_k[m,n]$, denoted by $\bar{\Delta}[m,n]$, resulting from a small change $\tilde{\Delta}[m,n]$ in $\tilde{f}_k[m,n]$. It can be shown, by generalizing the result given by Fienup (1980,1981b), that $\bar{\Delta}[m,n]$ and $\tilde{\Delta}[m,n]$ are related by

$$E\left\{\bar{\Delta}[m,n]\right\} = \frac{1}{2}\,\tilde{\Delta}[m,n]\,E\left\{\frac{M[p,q]}{|\tilde{F}_k[p,q]|}\right\} \tag{3.57}$$

where $E\{\cdot\}$ denotes the expected value. Thus a change in $\tilde{f}_k[m,n]$ tends to result in a similar (but scaled) change in $\bar{f}_k[m,n]$. It is intuitively reasonable that this should be so, since both $\tilde{f}_k[m,n]$ and $\bar{f}_k[m,n]$ possess identical spectral phase. It then follows that a change in $\tilde{f}_k[m,n]$ should be accompanied by a corresponding change in $\bar{f}_k[m,n]$, because of the dominance of phase in determining the form of an image (see §2.2).

Fienup (1980,1982) notes that, in light of the preceding discussion, it is useful to consider the first three steps of the error-reduction algorithm (i.e. (3.49a–c)) as a non-linear system, the input and output of which are $\tilde{f}_k[m,n]$ and $\bar{f}_k[m,n]$ respectively, as depicted in Figure 3.2. By doing so, algorithms can be devised which vary the input $\tilde{f}_k[m,n]$ in such a manner as to "persuade" the output $\bar{f}_k[m,n]$ to satisfy the image space constraints. Note that any $\bar{f}_k[m,n]$ which does satisfy the image space constraints is then a solution, since all $\bar{f}_k[m,n]$ satisfy (by definition) the Fourier space constraints.

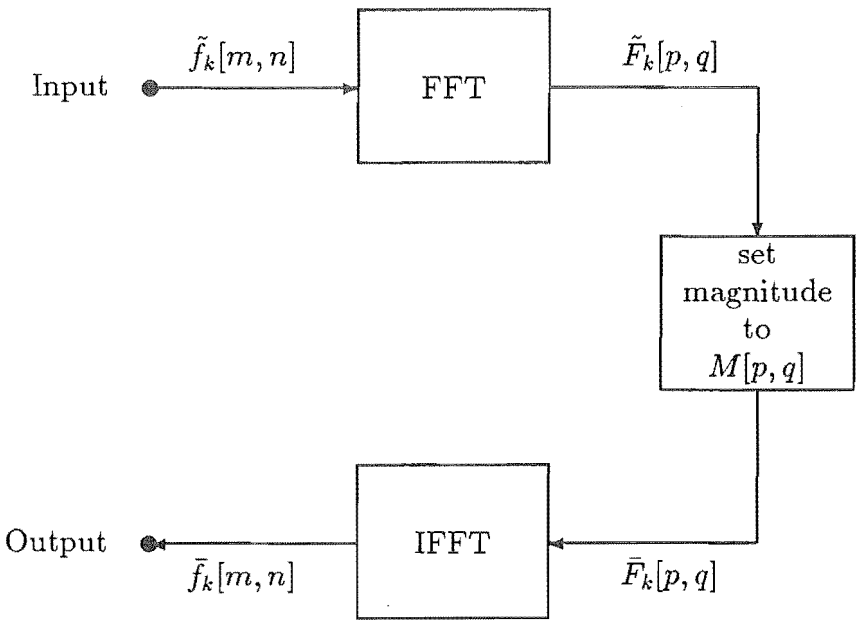


Figure 3.2 Depiction of the first three steps of the error-reduction algorithm (3.49a–c). By considering this process as a non-linear system, the input-output algorithms can be derived.

One such possibility is to change the input at any pixel for which the corresponding output pixel violates the constraints, but make no change to the input at

pixels for which the corresponding output pixel satisfies the constraints. The change in the input is made in proportion to, but in the opposite direction from, the corresponding output. The new input $\tilde{f}_{k+1}[m,n]$ is then

$$\tilde{f}_{k+1}[m,n] = \begin{cases} \tilde{f}_k[m,n] & \text{if } [m,n] \notin V_k \\ \tilde{f}_k[m,n] - \beta\, \bar{f}_k[m,n] & \text{if } [m,n] \in V_k \end{cases} \tag{3.58}$$

where β is a positive constant. The algorithm given by (3.49) with (3.49d) replaced by (3.58) is termed the *basic input-output algorithm* by Fienup (1982).

An alternative method of forming $\tilde{f}_{k+1}[m,n]$ is to combine the top line of (3.49d) with the bottom line of (3.58), yielding

$$\tilde{f}_{k+1}[m,n] = \begin{cases} \bar{f}_k[m,n] & \text{if } [m,n] \notin V_k \\ \tilde{f}_k[m,n] - \beta\, \bar{f}_k[m,n] & \text{if } [m,n] \in V_k \end{cases} \tag{3.59}$$

The algorithm given by (3.49), with (3.49d) replaced by (3.59), is known as Fienup's *hybrid input-output algorithm.* It has been demonstrated (Fienup, 1982) that input-output algorithms are generally more effective than the error-reduction algorithm, and that hybrid input-output is the most effective of the input-output algorithms yet devised.

It seems reasonable that the hybrid input-output algorithm ought to outperform both the error-reduction and basic input-output algorithms, for the following reasons. Consider first the operation of either the basic or hybrid input-output algorithm, in which a positivity constraint is enforced, at a particular pixel $[m',n']$. Suppose that $\bar{f}[m',n']$ has significant negative value. The corresponding pixel $\tilde{f}[m',n']$ in the next input is increased at each iteration, until $\bar{f}[m',n']$ eventually becomes positive. The error-reduction algorithm, in such a situation, simply sets $\tilde{f}[m',n'] = 0$ at each iteration. The input-output algorithms thus cannot stagnate in the same manner as the error-reduction algorithm.

Consider now the operation of the basic input-output algorithm with solely a support constraint. The image space constraints are evidently satisfied at any pixel of $\bar{f}[m,n]$ within the image mask, and thus none of the pixels of $\tilde{f}[m,n]$ which are within the image mask are ever modified, preventing the convergence of the algorithm. The hybrid input-output algorithm, on the other hand, allows the input to follow the output at pixels within the image mask, by setting $\tilde{f}[m,n] = \bar{f}[m,n]$.

Although hybrid input-output is the most effective of Fienup's algorithms, no detailed analysis of the algorithm has as yet been reported. Although the algorithm's worth has been proven by its performance, an analysis would be valuable, since it might suggest further improvements which could be made to the algorithm.

3.2.8.3 The Practical Application of Fienup's Algorithms

Poor results are usually obtained if Fienup's algorithms are carelessly applied to a particular phase problem. This section briefly describes some of the precautions which are essential if one is to obtain useful results from these algorithms. The majority of the techniques described here have been incorporated into a composite phase retrieval algorithm (Bates and Fright, 1983; Bates and Fright, 1984, §11,12;

Bates and Mnyama, 1986, §V; Fright, 1984, ch. 7), often referred to as the *Canterbury algorithm*. Further details regarding the application of Fienup's algorithms are given by Fienup and Wackerman (1986) and Dainty and Fienup (1987, §7.4B,C).

Before any iterative processing is begun, it is often advantageous to *preprocess* the measured data. This typically involves calculating the autocorrelation (by performing an inverse FFT on the intensity data), setting to zero of all parts of the autocorrelation which are negative or are deemed to be outside the autocorrelation support, and finally performing an FFT on the modified autocorrelation to yield the preprocessed intensity data (Bates and Mnyama, 1986, §V.A).

Fienup's algorithms require an estimate of the support of the image which is to be reconstructed, so that an image mask may be defined, as described in §3.2.8.1. The simplest image mask is a rectangle of half the linear dimensions of the smallest rectangle which encloses the autocorrelation, but more sophisticated methods of mask determination are available (Fienup *et al.*, 1982; Crimmins *et al.*, 1990). It should be noted that Fienup's algorithms tend to converge most rapidly when the image mask is of the smallest possible size (i.e. corresponds exactly with the support of the original image), especially if a positivity constraint is not employed. Fienup (1987) shows that his algorithms can reconstruct complex images which possess supports of special shapes (e.g. triangular, or consisting of separated parts). Lane (1987) also notes that they can be effective, although much less efficient, for images with rectangular support. Fienup and Wackerman (1986) suggest using a mask which is smaller than the correct support for the first few iterations, enlarging the mask to the size of the correct support for the majority of the iterations, then enlarging the mask even further for a final small number of iterations.

To begin the algorithm, an initial image estimate is required. This often simply comprises an image containing pseudo-random numbers within the image mask. Alternatively, a starting image may be computed via a crude phase estimate (see §3.2.5.2), which usually results (especially when the data are appreciably contaminated) in more rapid convergence of Fienup's algorithms than when a pseudo-random starting image is employed (Bates and Fright, 1983).

It has become standard practice, when applying Fienup's algorithms, to alternately perform a number of error-reduction iterations followed by a number of hybrid input-output iterations, which is known as *Fienup cycling* (Bates and Mnyama, 1986, §V.D). One advantage in so doing is that the value of the error E_k^{I} computed after a number of error-reduction iterations have been performed may correspond better with image quality than the value of E_k^{I} computed while hybrid input-output iterations are being performed (Fienup and Wackerman, 1986). However, Lane (1987) has demonstrated that, when reconstructing complex images, the performance of the hybrid input-output algorithm alone is superior to a combination of hybrid input-output and error-reduction.

When applying Fienup's algorithms, it must somehow be decided when to cease iterating. A simple strategy is to perform a pre-chosen number of iterations. Alternative strategies can be devised which monitor the value of the error E_k^{I} as the iterations proceed, which is known as *error watching*. The simplest error watching strategy involves ceasing iteration when E_k^{I} falls below some value E^0, determined from consideration of the contamination on the data (Bates and Mnyama, 1986, §IV.D;

Fienup and Wackerman, 1986). More sophisticated error watching strategies are discussed in Chapter 4.

If, when applying Fienup's algorithms, the error E_k^I fails to decrease significantly over a number of iterations, but is significantly greater than E^0, the algorithm is said to have *stagnated*. Various methods of overcoming stagnation are now discussed. One class of images for which Fienup's algorithms often stagnate are those which contain faint detail on a bright background, implying that the Fourier magnitude is dominated by a large central lobe. In such cases, it has been shown (Bates and Fright, 1983; Bates and Mnyama, 1986, §V.E) that the convergence of Fienup's algorithms is superior when operating upon $W(u,v)|F(u,v)|$ rather than on the given $|F(u,v)|$, where $W(u,v)$ is a positive function which appropriately reduces the central lobe of $|F(u,v)|$. After performing a number of iterations using the modified magnitude $W(u,v)|F(u,v)|$, a procedure known as *defogging*, further iterations are performed using $|F(u,v)|$.

Fienup and Wackerman (1986) describe three other stagnation mechanisms, and offer means of overcoming each of them. A mode of stagnation which can occur when the image mask is centro-symmetric (or nearly so) involves the appearance in the reconstructed image of two versions of the original image, one of which is rotated through 180°. This mode of stagnation can be overcome by performing a small number of iterations with a non-centro-symmetric (e.g. triangular) mask, followed by further iterations with the correct mask.

Another mode of stagnation arises when the reconstructed image is imperfectly located within the image mask. Because the fraction of the energy of the reconstructed image which violates the support constraint is small if the mask is misplaced by only a few pixels, the algorithm's convergence becomes very slow. This mode of stagnation can be overcome by simply enlarging the mask by a few pixels during the final iterations, as previously mentioned. Alternatively, the reconstructed image can be dynamically translated as the iterations proceed, such that the energy of the parts of the reconstructed image lying within the mask is maximized. Such translation can either be performed at every iteration, or only after a pre-chosen number of iterations, in which case it can be advantageous to temporarily enlarge the mask for a few iterations before effecting the translation procedure.

Yet another mode of stagnation involves the appearance of stripes across the reconstructed image, which extend outside the image mask. Fienup and Wackerman (1986) demonstrate two methods for overcoming this problem, involving appropriately combining a few differently striped reconstructed images.

Fienup's algorithms can be remarkably effective when applied with due regard to the precautions outlined above. It has already been mentioned that these algorithms can be employed to recover complex images (Fienup, 1987; Lane, 1987). Cederquist *et al.* (1988) have also demonstrated their effectiveness in the reconstruction of complex images from measured magnitude data. Furthermore, Fienup's algorithms are usefully robust in the presence of noise on the magnitude data (e.g. Fienup, 1978; Bates and Mnyama, 1986, §IV.F). Feldkamp and Fienup (1980) show that images reconstructed from contaminated magnitude data degrade gradually as the contamination level increases, and that the rms error of the reconstructed images is roughly proportional to the square root of the rms of the contamination. The behaviour

of the hybrid input-output algorithm when operating on significantly contaminated magnitude data is discussed further in Chapter 4.

3.2.8.4 Method of Generalized Projections

As explained in §3.1.1.2, the theory of projections onto convex sets cannot be directly applied to the phase problem, since the set of images with a prescribed Fourier transform magnitude is not a convex set. However, Levi and Stark (1984,1987) have generalized the projections onto convex sets algorithm, to cater for image reconstruction problems in which one or more of the constraint sets are non-convex. The resulting method is termed the *method of generalized projections.*

Useful results have only been derived for problems involving two constraint sets. This is not overly restrictive, however, since constraints can often be combined into a single set, for which a projection operator can still be defined (e.g. all image domain constraints can usually be incorporated into one constraint set, and all Fourier domain constraints incorporated into another). The generalized projection algorithm is given by

$$\hat{f}_{n+1} = T_2\, T_1\, \hat{f}_n \tag{3.60}$$

where $\hat{f}_n$ is the n^{th} image estimate, and T_1 and T_2 are defined by (3.8). Note that (3.60) is equivalent to (3.7) with only two constraint sets. For the phase problem, the constraint set C_1 is the set of images with a prescribed support, whereas C_2 is the set of images with a prescribed Fourier transform magnitude. Note that (3.60) reduces to Fienup's error-reduction algorithm (§3.2.8.1) when $\lambda_1 = \lambda_2 = 1$, where λ_1 and λ_2 are defined in (3.8).

A useful convergence measure is the *summed distance error* $J(f_n)$ as defined by

$$J(f_n) = \|P_1\, f_n - f_n\| + \|P_2\, f_n - f_n\| \tag{3.61}$$

which is simply the sum of the distances from f_n to C_1 and C_2. It can be shown (Levi and Stark, 1984) that

$$J(f_{n+1}) \leq J(T_1\, f_n) \leq J(f_n) \tag{3.62}$$

for certain values of λ_1 and λ_2 (which always include $\lambda_1 = \lambda_2 = 1$). Note that when $\lambda_1 = \lambda_2 = 1$, (3.62) is equivalent to the non-increasing error property of the error-reduction algorithm, as given by (3.52).

Unfortunately, the method of generalized projections often suffers stagnation problems, just as Fienup's error-reduction algorithm does. Levi and Stark (1984) attribute this to the existence of *traps*, which are fixed points of the operator $T_1 T_2$, but are not fixed points of both T_1 and T_2. However, Levi and Stark (1984) offer no suggestions as to how such traps may be avoided.

Çetin and Ansari (1988) have considered signal recovery algorithms in a Hilbert space $\mathcal{H}^*$, in which vector addition is defined as convolution. Scalar multiplication in $\mathcal{H}^*$ is also appropriately defined, so that the set of signals with a prescribed Fourier transform magnitude is a convex set in $\mathcal{H}^*$. Unfortunately, the set of signals with a prescribed support is not a convex set in $\mathcal{H}^*$, and so the theory of projections onto convex sets cannot be directly applied. However, Çetin and Ansari (1988) demonstrate that approximate solutions to the two-dimensional phase problem can

be obtained by making certain assumptions regarding the support of the image's cepstrum (because the set of images with a prescribed cepstral support is a convex set in $\mathcal{H}^*$). Such solutions may well be of use as starting estimates for other phase retrieval algorithms.

3.2.9 Optimization Methods

This section describes various phase retrieval techniques which employ some form of optimization method (see §3.1.2). Techniques using gradient methods are described in §3.2.9.1, whilst §3.2.9.2 details the application of simulated annealing to phase retrieval. Finally, phase retrieval by maximum entropy methods is mentioned in §3.3.4.

3.2.9.1 Gradient Methods

It has been mentioned, in §3.2.8.1, that the error-reduction algorithm is a special case of the steepest descent method described in §3.1.2.1. Fienup (1982) shows that the general form of the steepest descent method, as applied to the phase problem, involves replacing (3.49d) of the error-reduction algorithm by the following two steps:

$$\begin{aligned} f'_k[m,n] &= \tilde{f}_k[m,n] - s_k \nabla Q_k[m,n] \\ &= \tilde{f}_k[m,n] - 2s_k(\tilde{f}_k[m,n] - \bar{f}_k[m,n]) \end{aligned} \tag{3.63a}$$

$$\tilde{f}_{k+1}[m,n] = \begin{cases} f'_k[m,n] & \text{if } [m,n] \notin V_k \\ 0 & \text{if } [m,n] \in V_k \end{cases} \tag{3.63b}$$

where the gradient $\nabla Q_k[m,n]$ is defined by (3.54), and s_k denotes the k^{th} step size (the error-reduction algorithm is obtained by setting $s_k = \frac{1}{2}$). Note that (3.63a) effects a step in the steepest descent direction, whereas (3.63b) enforces the image space constraints.

Fienup (1982) also considers the application of conjugate gradient methods to the phase problem. This involves replacing (3.49d) of the error-reduction algorithm by

$$f'_k[m,n] = \tilde{f}_k[m,n] + s_k\, d_k[m,n] \tag{3.64}$$

followed by (3.63b), where $d_k[m,n]$ is the k^{th} search direction, given by

$$d_k[m,n] = \bar{f}_k[m,n] - \tilde{f}_k[m,n] + \frac{Q_k}{Q_{k-1}}\, d_{k-1}[m,n] \tag{3.65}$$

and Q_k denotes $Q\{\hat{f}[m,n]\}$ evaluated at $\hat{f}[m,n] = \tilde{f}_k[m,n]$. In the one example presented by Fienup (1982), the performance of the general steepest descent method, with fixed s_k satisfying $s_k > \frac{1}{2}$, is superior to that of the error-reduction algorithm. Also, the performance of the conjugate gradient method, with fixed s_k, appears to be superior to that of the steepest descent method.

The application of conjugate gradient methods to phase retrieval has also been investigated by Sasaki and Yamagami (1987). Their modification of Fienup's conjugate gradient method, which incorporates a positivity constraint by invocation of the Kuhn-Tucker theorem, is shown to converge faster than Fienup's method.

Sasaki and Yamagami (1987) also propose an alternative method, to that described in §3.2.8.1, of posing the phase problem as a minimization problem. To do so, the image estimate $\hat{f}[m,n]$ is expressed as

$$\hat{f}[m,n] \longleftrightarrow \hat{F}[p,q] = M[p,q]\exp(i\,\hat{\Phi}[p,q]) \tag{3.66}$$

Instead of seeking that $\hat{f}[m,n]$ which minimizes $Q\{\hat{f}[m,n]\}$, a $\hat{\Phi}[p,q]$ is sought which minimizes the cost function $Q\{\hat{\Phi}[p,q]\}$, as defined by

$$Q\{\hat{\Phi}[p,q]\} = \sum_{m,n\,:\,mask[m,n]=0}\sum \left(\hat{f}[m,n]\right)^2 \tag{3.67}$$

where the mask buffer $mask[m,n]$ is zero outside the estimated support of $f[m,n]$, as described in §3.2.8.1. Sasaki and Yamagami (1987) minimize $Q\{\hat{\Phi}[p,q]\}$ by the steepest descent method, whilst applying a positivity constraint. They also demonstrate a procedure in which $Q\{\hat{f}[m,n]\}$ and $Q\{\hat{\Phi}[p,q]\}$ are minimized alternately. However, the results presented are for small, simple images, and it is not clear whether these techniques are effective for larger images.

Nieto-Vesperinas (1986) has also considered the application of gradient methods to phase retrieval, namely a modified Gauss-Newton method (Dennis and Scnabel, 1983, ch. 10), which has the advantage of quadratic convergence. To apply such techniques, the phase problem is expressed as the problem of finding that $\hat{f}[m,n]$, for which

$$\hat{f}[m,n] = 0 \text{ for } m,n \,:\, mask[m,n] = 0 \tag{3.68}$$

which minimizes the cost function $Q'\{\hat{f}[m,n]\}$, as defined by

$$Q'\{\hat{f}[m,n]\} = \sum_m \sum_n \left(a[m,n] - \widehat{ff}[m,n]\right)^2 \tag{3.69}$$

where $a[m,n]$ is the measured autocorrelation of $f[m,n]$, and $\widehat{ff}[m,n]$ is the autocorrelation of $\hat{f}[m,n]$. If it is assumed that the values of $mask[m,n]$ are zero outside a $N{\times}N$ pixel rectangle, then only N^2 values of $\hat{f}[m,n]$ must be determined, and $\widehat{ff}[m,n]$ can then only possess non-zero values within a $(2N-1){\times}(2N-1)$ pixel rectangle. If $a[m,n]$ only possesses non-zero values within the aforementioned $(2N-1){\times}(2N-1)$ pixel rectangle, then the summations in (3.69) need only cover values of m and n for which $[m,n]$ lies within this rectangle. However, since any autocorrelation is centro-symmetric, the summations in (3.69) actually need only cover $N^2+(N-1)^2$ appropriately selected pixels, rather than the full $(2N-1)^2$ pixels.

Although the method of Nieto-Vesperinas (1986) enjoys rapid convergence and robustness in the presence of noise on the data, it generally converges to a local, rather than a global, minimum of $Q'\{\hat{f}[m,n]\}$. Despite the number of local minima encountered being shown to increase rapidly with image size (thereby rendering this method ineffective for all but the smallest images), Nieto-Vesperinas (1986) points out that this study provides valuable insight into the nature of the phase problem. Clearly, no simple-minded algorithm which attempts to solve the phase problem by any method which is equivalent to minimizing $Q'\{\hat{f}[m,n]\}$ can be expected to perform successfully.

3.2.9.2 Simulated Annealing Method

Because the cost function $Q'\{\hat{f}[m,n]\}$, as defined in (3.69), possesses multiple minima, a global optimization algorithm is required if the phase problem is to be solved by minimization of $Q'\{\hat{f}[m,n]\}$. Nieto-Vesperinas and Mendez (1986) demonstrate that simulated annealing, a global optimization algorithm which has been described in §3.1.2.2, can be usefully applied to the phase problem. An improved algorithm for effecting phase retrieval by simulated annealing was reported by Nieto-Vesperinas *et al.* (1988), and the application of this algorithm to image reconstruction from simulated stellar speckle interferometry data has been described by Navarro *et al.* (1989).

The most important aspects of the simulated annealing phase retrieval algorithm of Nieto-Vesperinas *et al.* (1988) are now described. A pixel of the image estimate $\hat{f}[m,n]$ is perturbed by an amount p, which is pseudo-randomly distributed in the range $[-\alpha,\alpha]$, with α being termed the *scale of perturbation*. The change in the cost function resulting from this perturbation is then calculated, and the perturbation is accepted or rejected according to the criteria described in §3.1.2.2, under the control of the temperature parameter T. One *image scan* constitutes sequentially perturbing each pixel of $\hat{f}[m,n]$ in this manner. Image scans are repeated until *equilibrium* is reached. This is deemed to have occurred when the average number of perturbations increasing $Q'\{\hat{f}[m,n]\}$ is sufficiently close to the average number of perturbations decreasing $Q'\{\hat{f}[m,n]\}$. Once equilibrium is reached, the values of both T and α are decreased according to various prescriptions. The above procedure, which constitutes one *cycle*, is then repeated a number of times.

To reduce the computational burden involved in computing $Q'\{\hat{f}[m,n]\}$ following each perturbation, an efficient means of calculating $\widehat{ff}[m,n]$ is employed. This is conveniently described with the aid of the continuous image notation. Consider a perturbation p made at a point (x',y') to the k^{th} image estimate $\hat{f}_k(x,y)$. The $(k+1)^{\text{th}}$ autocorrelation $\widehat{ff}_{k+1}(x,y)$ can then be efficiently computed from $\widehat{ff}_k(x,y)$, since

$$\begin{aligned}\widehat{ff}_{k+1}(x,y) &= \left[\hat{f}_k(x,y)+p\,\delta(x-x',y-y')\right] * \left[\hat{f}_k(x,y)+p\,\delta(x-x',y-y')\right] \quad (3.70)\\ &= \widehat{ff}_k(x,y)+p\,\hat{f}_k(x+x',y+y')+p\,\hat{f}_k(-x-x',-y-y')+p^2\delta(x,y)\end{aligned}$$

Nieto-Vesperinas *et al.* (1988) conclude that their simulated annealing algorithm is robust in the presence of noisy data, and flexible as regards the incorporation of *a priori* constraints, but that it is undeniably computationally intensive. They point out that the simulated annealing method is often capable of considerably improving images reconstructed by Fienup's algorithms, and that the appropriate combination of Fienup's algorithms with simulated annealing is less computationally demanding than employing simulated annealing alone. An annealing scheme mentioned by Navarro *et al.* (1989), shows particular promise for recovering faithful images from significantly contaminated data. This involves gradually reducing T to some prescribed value T_0, then fixing $T=T_0$ and averaging the image estimates obtained at the end of each of a number of image scans.

It is important to note that the simulated annealing algorithm and Fienup's hybrid input-output algorithm (§3.2.8.2) are the only algorithms which, to date, have been demonstrated as being capable of effecting the phase retrieval of general images of

reasonable size from contaminated data. An alternative method of applying simulated annealing to the phase problem is described in §5.4.

3.2.9.3 Maximum Entropy Method

The application of maximum entropy methods to the phase problem has been outlined by Gull and Daniell (1978) and Gull and Skilling (1984). The essence of a more detailed presentation, given by Bryan and Skilling (1986), is now described.

As discussed in general terms in §3.1.2.3, a measure of the discrepancy between the measured intensity data $I[p,q]$ (which is assumed to have been sampled on an $M \times M$ grid) and the ideal intensity data $|\hat{F}[p,q]|^2$ that would result from the image estimate $\hat{f}[m,n]$ (where $\hat{F}[p,q]$ is the DFT of $\hat{f}[m,n]$) is required. Such a measure is given by

$$\chi^2\{\hat{f}[m,n]\} = \sum_{p=1}^{M}\sum_{q=1}^{M} \frac{\left(|\hat{F}[p,q]|^2 - I[p,q]\right)^2}{\sigma[p,q]^2} \tag{3.71}$$

where $\sigma[p,q]$ is the estimated standard deviation of the contamination on $I[p,q]$.

Solving the phase problem by the maximum entropy method then involves finding that $\hat{f}[m,n]$ which maximizes the entropy $S\{\hat{f}[m,n]\}$, as defined by (3.1.2.3), subject to the constraint that $\chi^2\{\hat{f}[m,n]\} = M^2$. For situations in which the data constraints are linear combinations of the unknown $\hat{f}[m,n]$, as for the deconvolution problem, it can be shown (Gull and Daniell, 1978) that the entropy $S\{\hat{f}[m,n]\}$ possesses a single extremum, which is a maximum. Unfortunately, the data constraints for the phase problem are not of this form, implying that $S\{\hat{f}[m,n]\}$ possesses multiple extrema in general. Sophisticated algorithms are thus required to effect phase retrieval by maximum entropy methods.

Bryan and Skilling's (1986) algorithm operates by attempting to follow the loci of maxima of $S\{\hat{f}[m,n]\}$ on surfaces of constant $\chi^2\{\hat{f}[m,n]\}$. The value of $\chi^2\{\hat{f}[m,n]\}$ is gradually decreased until it reaches M^2, at which stage the algorithm has necessarily found an entropy maximum which is consistent with the data constraints. However, because of the complicated topology of the entropy surface, there is no way of determining if the algorithm has located a local or global entropy maximum. It is possible to run the algorithm a number of times, each time starting at a different point, and retain the result for which the entropy is maximum. As well as being difficult to implement, this algorithm is rather computationally intensive. The examples presented by Bryan and Skilling (1986) suggest that maximum entropy phase retrieval can be effective when reconstructing images comprising isolated points, but ineffective for images of extended objects. This is in accord with the experience of researchers in other areas (e.g. Nityananda and Narayan, 1982).

Narayan (1987) examines the topology of the entropy surface, and considers the implications that this has upon general maximum entropy phase retrieval algorithms. It is also worth mentioning that maximum entropy methods have been applied to the crystallographic phase problem. The work of Bricogne (1984) and Narayan *et al.* (1983) may also be of relevance to maximum entropy solutions to Fourier phase problems.

3.3 Methods for Solving Blind Deconvolution Problems

This section describes various methods for solving blind deconvolution problems. Because the blind deconvolution problem has only received significant attention over the last few years, as opposed to decades for the phase problem, it is not at all surprising that there are far fewer techniques available for blind deconvolution than there are for phase retrieval. However, since the phase problem is a special case of the blind deconvolution problem (as explained in §2.6), many of the techniques devised for phase retrieval may be applicable, after appropriate modification, to blind deconvolution. Indeed, some of the techniques described in this section have been devised on this basis.

Before describing any methods for effecting blind deconvolution, it is first necessary to briefly describe some conventional deconvolution methods, which is done in §3.3.1. Blind deconvolution via zero-sheets is discussed in §3.3.2, whereas iterative blind deconvolution algorithms are detailed in §3.3.3. Finally, blind deconvolution by maximum entropy techniques is described in §3.3.4. Two new blind deconvolution techniques are later described in Chapters 6 and 7.

3.3.1 Conventional Deconvolution Methods

The conventional deconvolution problem, which has already been mentioned in §2.6, has been considered by many authors (e.g. Sondhi, 1972; Andrews and Hunt, 1977; Bates and McDonnell, 1986, ch. 3), and a great number of techniques have been developed for solving it. However, the purposes of the present discussion are served by neglecting all but two such techniques here.

A simple-minded approach to solving the conventional deconvolution problem as posed in §2.6, in which the contaminated convolution $g(\vec{x})$ is given by

$$g(\vec{x}) = f(\vec{x}) \odot h(\vec{x}) + c(\vec{x}) \tag{3.72}$$

is to compute an estimate $\hat{f}(\vec{x})$ of $f(\vec{x})$ by

$$\hat{f}(\vec{x}) \longleftrightarrow \hat{F}(\vec{u}) = \frac{G(\vec{u})}{\hat{H}(\vec{u})} \tag{3.73}$$

where $\hat{h}(\vec{x})$ is an estimate of $h(\vec{x})$. This technique, which is known as *inverse filtering*, is seldom employed in practice for the following reason. From (3.73) and (3.72), it is evident that

$$\hat{F}(\vec{u}) = F(\vec{u}) \frac{H(\vec{u})}{\hat{H}(\vec{u})} + \frac{C(\vec{u})}{\hat{H}(\vec{u})} \tag{3.74}$$

Now if $|C(\vec{u})| \ll |\hat{H}(\vec{u})|$ throughout Fourier space, and if $\hat{H}(\vec{u})$ is sufficiently similar to $H(\vec{u})$, then $\hat{F}(\vec{u})$ is generally a useful estimate of $F(\vec{u})$. However, it usually happens that $|C(\vec{u})| \gg |\hat{H}(\vec{u})|$ within some parts of Fourier space, in which case these spatial frequencies dominate $\hat{f}(\vec{x})$. The artefacts which thereby appear in $\hat{f}(\vec{x})$ are known as *ringing*, which are typically so severe that $\hat{f}(\vec{x})$ is of no use as an estimate of $f(\vec{x})$.

The problems associated with the inverse filter can be overcome by the use of a *Wiener filter* $W(\vec{u})$, as defined by

$$W(\vec{u}) = \frac{H^*(\vec{u})}{|H(\vec{u})|^2 + \Phi(\vec{u})} \tag{3.75}$$

where $\Phi(\vec{u})$ is an estimate of the noise to signal ratio, as given by

$$\Phi(\vec{u}) = \frac{\mathrm{En}\{|C(\vec{u})|^2\}}{\mathrm{En}\{|F(\vec{u})|^2\}} \tag{3.76}$$

However, since the quantities on RHS (3.76) are usually unknown, it is common practice to simply assign a constant value to $\Phi(\vec{u})$. To apply the Wiener filter, $\hat{f}(\vec{x})$ is computed as

$$\hat{f}(\vec{x}) \longleftrightarrow \hat{F}(\vec{u}) = G(\vec{u})\, W(\vec{u}) \tag{3.77}$$

Although the Wiener filter can be shown to be optimum in certain circumstances (Andrews and Hunt, 1977, ch. 8), it is more useful (Bates *et al.*, 1984) to look upon the Wiener filter as a means of realizing a filter with response $1/H(\vec{u})$ (i.e. an inverse filter) when $H(\vec{u})$ is large and zero when $H(\vec{u}) = 0$, with smooth transitions (conditioned by $\Phi(\vec{u})$) for $H(\vec{u})$ between these extreme values (Bates and McDonnell, 1986, §16).

3.3.2 Complex Zero Based Methods

Lane and Bates (1987a) present a blind deconvolution method based on zero-sheets (see §2.4.2), and have demonstrated its effectiveness by blindly deconvolving two 32×32 pixel complex images. Their method is essentially identical to that of Lane *et al.* (1987) for phase retrieval via zero-sheets, which is described in §3.2.7.1. Unfortunately, blind deconvolution by zero-sheets is presently impractical when (as is always true in practice) the available data are contaminated, for the reasons stated in §3.2.7.

3.3.3 Iterative Methods

Three different iterative blind deconvolution techniques have been reported to date. That of Lane and Bates (1987b), in which phase retrieval is initially invoked, is described in §3.3.3.1. The iterative blind deconvolution algorithms of Ayers and Dainty (1988) and Davey *et al.* (1989) are detailed in §3.3.3.2.

3.3.3.1 Phase Based Method

The blind deconvolution technique described in this section, which is here called the *phase-based blind deconvolution algorithm*, was initially suggested by Lane and Bates (1987b), and demonstrated by Bates and Lane (1987), Bates and Davey (1987a) and Lane (1988, §6.3). This technique exploits the non-uniqueness of solution encountered when attempting to reconstruct an image from the magnitude of the Fourier transform of a convolution, as mentioned in §2.4.2.

Consider a discrete two-dimensional contaminated convolution $g[m,n]$, as given by

$$g[m,n] = f[m,n] \odot h[m,n] + c[m,n] \tag{3.78}$$

If it is assumed that $c[m,n] = 0$ in (3.78), then it is evident that $G[p,q]$ and the three quantities $G_2[p,q]$, $G_3[p,q]$ and $G_4[p,q]$, as defined by

$$G_2[p,q] = F^*[p,q]\,H[p,q] \tag{3.79a}$$
$$G_3[p,q] = F[p,q]\,H^*[p,q] \tag{3.79b}$$
$$G_4[p,q] = F^*[p,q]\,H^*[p,q] \tag{3.79c}$$

all possess the same magnitude. However, since $g[m,n]$ and $g_4[m,n]$ possess the same image-form (§2.1), as do $g_2[m,n]$ and $g_3[m,n]$, there are only two image-forms compatible with $|G[p,q]|$.

The first step in the phase-based blind deconvolution algorithm involves employing Fienup's algorithms to reconstruct an image $g'[m,n]$ from $|G[p,q]|$. If $g'[m,n]$ does not possess a significantly different image-form from that of $g[m,n]$, then this step is repeated, each time with a different starting image, until it does so. It then follows that either $g'[m,n] = g_2[m,n]$ or $g'[m,n] = g_3[m,n]$, but there is no way of knowing which equality applies. The next step is to compute the phase sum PS and phase difference PD as defined by

$$PS = \mathrm{Ph}\{G[p,q]\} + \mathrm{Ph}\{G'[p,q]\} \tag{3.80}$$
$$PD = \mathrm{Ph}\{G[p,q]\} - \mathrm{Ph}\{G'[p,q]\} \tag{3.81}$$

From (3.79b,c), it is evident that

$$\begin{array}{l} PS = 2\,\mathrm{Ph}\{H[p,q]\} \text{ and } PD = 2\,\mathrm{Ph}\{F[p,q]\} \quad \text{if } G'[p,q] = G_2[p,q] \\ PS = 2\,\mathrm{Ph}\{F[p,q]\} \text{ and } PD = 2\,\mathrm{Ph}\{H[p,q]\} \quad \text{if } G'[p,q] = G_3[p,q] \end{array} \tag{3.82}$$

implying that PS and PD contain the phase of $F[p,q]$ and of $H[p,q]$ (or vice-versa). However, since $\mathrm{Ph}\{G[p,q]\}$ and $\mathrm{Ph}\{G'[p,q]\}$ are computed (as opposed to being determined analytically), their values are only available modulo 2π. Thus it is only possible to deduce $\mathrm{Ph}\{F[p,q]\}$ mod π and $\mathrm{Ph}\{H[p,q]\}$ mod π from (3.82).

Lane and Bates (1987b) present and demonstrate an algorithm which effects the reconstruction of an image from its modulo π Fourier phase, which is based on the Hayes-Lim-Oppenheim magnitude retrieval algorithm (§3.1.1.1). Examination of this algorithm reveals that it is closely related to the projections onto convex sets algorithm described in §3.1.1.2 (specifically, the constraint sets are convex, but the Fourier domain constraint is not a projection). The final step of the phase-based blind deconvolution algorithm involves the independent reconstruction of $F[p,q]$ and $H[p,q]$ from $\mathrm{Ph}\{F[p,q]\}$ mod π and $\mathrm{Ph}\{H[p,q]\}$ mod π. It has been suggested (Bates and Lane, 1987; Lane, 1988, §6.3) that a more effective strategy is to recover $F[p,q]$ and $H[p,q]$ simultaneously, exploiting the extra information contained in $G[p,q]$, namely $|F[p,q]||H[p,q]|$ and $(\mathrm{Ph}\{F[p,q]\} + \mathrm{Ph}\{H[p,q]\})$ mod 2π.

3.3.3.2 Iterative Blind Deconvolution Method

The blind deconvolution algorithms presented by Ayers and Dainty (1988) and Davey *et al.* (1989) are both special cases of a general blind deconvolution technique, which is here called the *iterative blind deconvolution algorithm.* Figure 3.3 depicts the basic

loop of the iterative blind deconvolution algorithm, the operation of which is now described in terms of the contaminated convolution $g[m,n]$ as defined by (3.78). It is convenient to begin the algorithm with an estimate $\tilde{f}[m,n]$ of $f[m,n]$. The algorithm can of course be started at any point in the loop. An estimate of $h[m,n]$, denoted by $\bar{h}[m,n]$, is then obtained by deconvolving $\tilde{f}[m,n]$ from $g[m,n]$. Image space constraints are enforced upon $\bar{h}[m,n]$ to produce $\tilde{h}[m,n]$, which is then deconvolved from $g[m,n]$ to give $\bar{f}[m,n]$. Finally, image space constraints are enforced upon $\bar{f}[m,n]$ yielding a new $\tilde{f}[m,n]$, thereby establishing an iterative loop.

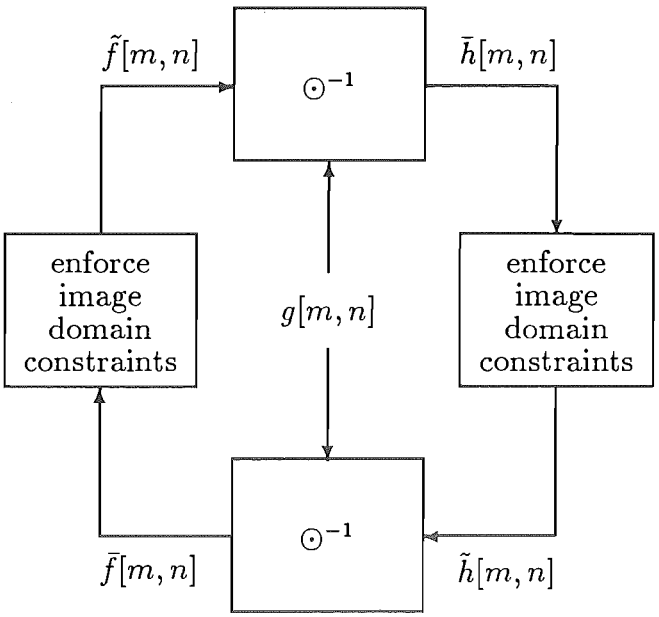


Figure 3.3 The basic loop of the iterative blind deconvolution algorithm.

The steps of the algorithm can be written as

$$\bar{h}_k[m,n] = g[m,n] \odot^{-1} \tilde{f}_k[m,n] \tag{3.83a}$$

$$\tilde{h}_k[m,n] = \begin{cases} \bar{h}_k[m,n] & \text{if } [m,n] \notin V_k^h \\ 0 & \text{if } [m,n] \in V_k^h \end{cases} \tag{3.83b}$$

$$\bar{f}_k[m,n] = g[m,n] \odot^{-1} \tilde{h}_k[m,n] \tag{3.83c}$$

$$\tilde{f}_{k+1}[m,n] = \begin{cases} \bar{f}_k[m,n] & \text{if } [m,n] \notin V_k^f \\ 0 & \text{if } [m,n] \in V_k^f \end{cases} \tag{3.83d}$$

where variables subscripted with a k denote quantities at the k^{th} iteration, and where V_k^f and V_k^h comprise the sets of pixels at which $\bar{f}_k[m,n]$ and $\bar{h}_k[m,n]$ violate their respective image space constraints. The algorithms of Ayers and Dainty (1988) and Davey *et al.* (1989) differ only in the particular image space constraints applied, and in the manner in which the deconvolutions are performed.

The sole image space constraint applied in Ayers and Dainty's (1988) algorithm is positivity. The deconvolution of $\tilde{h}[m,n]$ from $g[m,n]$, which yields $\bar{f}[m,n]$,

is implemented in Fourier space by

$$\bar{F}_k[p,q] = \begin{cases} \tilde{F}_k[p,q] & \text{if } |G[p,q]| \leq cl \\ (1-\beta)\tilde{F}_k[p,q] + \beta\dfrac{G[p,q]}{\tilde{H}_k[p,q]} & \text{if } |G[p,q]| > cl \text{ and } |\tilde{H}_k[p,q]| \geq |G[p,q]| \\ \dfrac{1}{\dfrac{1-\beta}{\tilde{F}_k[p,q]} + \dfrac{\beta\tilde{H}_k[p,q]}{G[p,q]}} & \text{if } |G[p,q]| > cl \text{ and } |\tilde{H}_k[p,q]| < |G[p,q]| \end{cases} \tag{3.84}$$

where cl is an estimate of the contamination level of $G[p,q]$, and β is a constant satisfying $0 < \beta \leq 1$. The deconvolution of $\tilde{f}[m,n]$ from $g[m,n]$ is effected by replacing $\bar{F}_k[p,q]$, $\tilde{F}_k[p,q]$ and $\tilde{H}_k[p,q]$ in (3.84) by $\bar{H}_k[p,q]$, $\tilde{H}_{k-1}[p,q]$ and $\tilde{F}_k[p,q]$ respectively. The rationale behind this method of calculating $\bar{F}_k[p,q]$ is now explained. If at some particular $[p,q]$, namely $[p',q']$, $|G[p',q']| < cl$, then $G[p',q']$ can provide no useful information regarding $F[p',q']$, and so $\tilde{F}_k[p',q']$ is the best available estimate of $F[p',q']$. But, if $|G[p',q']| > cl$, then both $\tilde{F}_k[p',q']$ and $G[p',q']/\tilde{H}_k[p',q']$ provide useful estimates of $F[p',q']$. This suggests that the average (weighted by β) of $\tilde{F}_k[p',q']$ and $G[p',q']/\tilde{H}_k[p',q']$ is a useful estimate of $F[p',q']$. However, if $|G[p',q']| > |\tilde{H}_k[p',q']|$, then such an average is dominated by $G[p',q']/\tilde{H}_k[p',q']$, in which case a weighted geometric average of $\tilde{F}_k[p',q']$ and $G[p',q']/\tilde{H}_k[p',q']$ is more sensible.

The image space constraints applied in the algorithm of Davey *et al.* (1989) include both support and (if appropriate) positivity. However, care must be exercised when applying this algorithm, to ensure that both the size and placement of the supports imposed upon $\bar{f}[m,n]$ and $\bar{h}[m,n]$ are consistent with those of the support of $g[m,n]$ (Davey, 1989, §7.4.1.1). The deconvolution of $\tilde{h}[m,n]$ from $g[m,n]$ is effected by a modified Wiener filter, as

$$\bar{F}_k[p,q] = \frac{G[p,q]\,(\tilde{H}_k[p,q])^*}{|\tilde{H}_k[p,q]|^2 + \Phi/|\tilde{H}_k[p,q]|^\alpha} \tag{3.85}$$

where Φ and α are constants (note that (3.85) reduces to a standard Wiener filter when $\alpha = 0$).

Davey (1989, §7.4) has demonstrated the superiority of Davey *et al.*'s (1989) algorithm over that of Ayers and Dainty (1988) when blindly deconvolving contaminated positive images. This is to be expected, since the imposition of a support constraint places strong restrictions on the form of the reconstructed images. Furthermore, Davey *et al.* (1989) also demonstrate that their algorithm is capable of blindly deconvolving a contaminated complex image.

3.3.4 Maximum Entropy Method

The application of maximum entropy methods to the blind deconvolution problem was first reported (albeit somewhat briefly) by Gull and Skilling (1984). They demonstrate the blind deconvolution of a binary image, comprising a number of letters, which had been blurred by an L-shaped psf. Their approach involves seeking estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ of the image and psf respectively, which maximize the joint entropy $S\{\hat{f}[m,n], \hat{h}[m,n]\}$ as defined by

$$S\{\hat{f}[m,n], \hat{h}[m,n]\} = \theta S\{\hat{f}[m,n]\} + (1-\theta)S\{\hat{h}[m,n]\}, \quad 0 \leq \theta \leq 1 \tag{3.86}$$

subject to the constraint that $\hat{f}[m,n] \odot \hat{h}[m,n]$ adequately agrees with the measured convolution data $g[m,n]$. $S\{\hat{f}[m,n]\}$ and $S\{\hat{h}[m,n]\}$ denote the entropy of $\hat{f}[m,n]$ and $\hat{h}[m,n]$ respectively, as defined by (3.1.2.3). A chi-squared statistic, similar to (3.14), is typically employed to implement the constraint that $\hat{f}[m,n] \odot \hat{h}[m,n]$ agrees with $g[m,n]$. Gull and Skilling (1984) demonstrate the effectiveness of their maximum entropy method by blindly deconvolving the convolution of two binary images.

The application of maximum entropy methods to the blind deconvolution problem has also been reported by Hildebrandt (1987) and Newman and Hildebrandt (1987), who provide details regarding the implementation of the method outlined by Gull and Skilling (1984), and the problems encountered in doing so.

Thus although it has been shown that blind deconvolution can be effected by maximum entropy methods, the results presented to date have involved only rather simple images. Furthermore, maximum entropy blind deconvolution techniques may only be effective for peaked images on a flat background (unless the contamination on the data is minimal), for the reasons given by Nityananda and Narayan (1982). On the other hand, some form of regularization method, but not necessarily maximum entropy, is probably essential if one wishes to obtain objective images from appreciably contaminated convolution data, because of the ill-posed nature of the blind deconvolution problem.

Chapter 4

Phase Retrieval From Contaminated Magnitude Data

When presented with a practical phase problem, the measured magnitude data is always contaminated, for the reasons stated in §1.1. Moreover, the level of contamination is significant in many situations. The hybrid input-output algorithm, as described in §3.2.8.2, is the most effective phase retrieval algorithm presently available. This chapter investigates the effect of contaminated magnitude data on the performance of the hybrid input-output algorithm.

The motivation for this study is discussed further in §4.1. In §4.2 it is demonstrated that the behaviour of the hybrid input-output algorithm is somewhat erratic when it operates upon significantly contaminated magnitude data. An attempt to improve the behaviour of the hybrid input-output algorithm by incorporating a threshold is described in §4.3. A new phase retrieval strategy, in which a number of appropriately selected reconstructed images are averaged to produce an enhanced image, is described in §4.4 and illustrated further in §4.5. This strategy has been reported by McCallum and Bates (1989). Finally, in §4.6, the techniques introduced in this chapter are assessed, and possible improvements to them are suggested.

4.1 Motivation

There is now little doubt amongst the image processing community that multidimensional Fourier phase problems always (except in contrived situations) possess unique solutions. This awareness is due to both the theoretical uniqueness arguments, which have been summarized in §2.4, and the many demonstrations of unique reconstructions obtained by the methods described in §3.2 (e.g. Fienup, 1983b; Bates and Mnyama, 1986). Furthermore, it is now clear that Fienup's algorithms (§3.2.8) are capable of recovering the phases of images exhibiting significant detail, provided that the precautions discussed in §3.2.8.3 are heeded. However, before Fienup's algorithms can be routinely employed in any particular application, automatic strategies must be devised in which the "best" reconstruction is computed with no (or as little as possible) intervention on the part of the user.

Any measured data which correspond to the magnitude of the Fourier transform of some quantity (which is here often referred to as simply the Fourier magni-

tude of that quantity) are unavoidably contaminated, for the reasons given in §1.1. However, various authors (e.g. Fienup, 1978; Bates and Mnyama, 1986, §IV.F) have demonstrated that Fienup's algorithms can even produce acceptable reconstructions when operating upon contaminated magnitude data. Furthermore, the faithfulness of the reconstructions tends to decrease only gradually as the contamination level increases. Feldkamp and Fienup (1980) have examined the relationship between the *squared reconstruction error* E^{R}, a measure of the discrepancy between the original and reconstructed images which is precisely defined in §4.2, and the *squared magnitude error* E^{M}, a measure of the discrepancy between the true magnitude $|F[p,q]|$ and the estimated magnitude $M[p,q]$, as given by

$$E^{\mathrm{M}} = \sum_p \sum_q (M[p,q] - |F[p,q]|)^2 \Big/ \sum_p \sum_q |F[p,q]|^2 \tag{4.1}$$

Feldkamp and Fienup (1980) show empirically that the reconstructions computed by Fienup's algorithms usually satisfy

$$E^{\mathrm{R}} \approx \sqrt{E^{\mathrm{M}}} \tag{4.2}$$

Various potential applications of phase retrieval exist in which the contamination on the Fourier magnitude data is severe. However, there is little information in the literature regarding either the behaviour or the performance of iterative phase retrieval algorithms when operating upon severely contaminated data. Furthermore, all reported applications of Fienup's algorithms to contaminated magnitude data (e.g. Fienup, 1978; Bates and Mnyama, 1986, §IV.F) have involved a combination of the hybrid input-output and error-reduction algorithms. Since Lane (1987) has shown that the convergence of hybrid input-output alone is usually superior to that of a combination of hybrid input-output and error-reduction, it seems worthwhile to investigate the sole use of hybrid input-output with contaminated magnitude data.

The behaviour and performance of the hybrid input-output algorithm when operating on severely contaminated data are detailed in §4.2. Based on the findings of §4.2, subsequent sections describe attempts at devising automatic strategies for computing useful reconstructions from severely contaminated magnitude data.

4.2 Behaviour of the Hybrid Input-Output Algorithm with Contaminated Data

The hybrid input-output algorithm has been described in detail in §3.2.8.2. Recall that the image error E_k^{I}, as defined by (3.50), is the fraction of the energy of $\bar{f}_k[m,n]$ (the estimate of $f[m,n]$ at the k^{th} iteration) which violates the image space constraints, and thus provides an indication of the algorithm's convergence. Monitoring the value of E_k^{I} as the iterations proceed, in order to estimate how well the algorithm has converged, is known as *error watching*. It is demonstrated in this section that the behaviour of the hybrid input-output algorithm is rather erratic when it operates upon severely contaminated magnitude data. Because of this erratic behaviour, simple error watching strategies that are effective when applying the hybrid

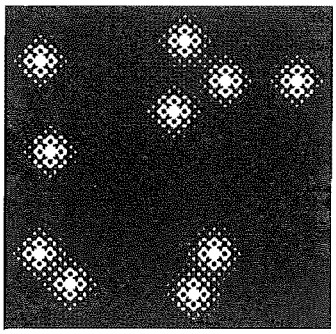


Figure 4.1 Central 32×32 pixel region of original 64×64 pixel image $f[m,n]$. This image is called the first original image.

input-output algorithm to uncontaminated (or slightly contaminated) data tend to be ineffective when the data are appreciably contaminated.

When the hybrid input-output algorithm operates on uncontaminated data, E_k^{I} usually falls steadily, although not necessarily monotonically, as the iterations proceed. Figure 4.2a shows the variation of E_k^{I} with k, henceforth called the *error curve*, obtained when the hybrid input-output algorithm operates on the uncontaminated Fourier magnitude of the image shown in Figure 4.1. Note that all of the images computed throughout this chapter comprise 64×64 pixels, but only the central 32×32 pixel region of any image is ever displayed. For this example, and for all other examples in this chapter unless it is stated otherwise, the algorithm is started with a pseudo-random initial image, then run for K iterations with $\beta = 0.5$, with both positivity and support constraints being enforced. After approximately 15 iterations, the reconstructed image $\bar{f}_k[m,n]$ is visually indistinguishable from the original image $f[m,n]$, although E_k^{I} continues to fall with further iterations. Thus, when applying the hybrid input-output to uncontaminated (or slightly contaminated) data, an appropriate error watching strategy is simply to iterate until E_k^{I} falls below some pre-set level.

When the magnitude data are significantly contaminated, the behaviour of the hybrid input-output algorithm is markedly different to that just described. Given the contaminated magnitude $M[p,q]$ of an image $f[m,n]$, there usually does not exist any estimate $\hat{f}[m,n]$ of $f[m,n]$ which both satisfies the appropriate image space constraints and has Fourier magnitude $M[p,q]$. Consequently, one cannot expect to be able to drive E_k^{I} below any arbitrarily small level. Instead, there is always some value, which depends on both the contamination level and the particular image space constraints being employed, below which E_k^{I} can never fall (Bates and Mnyama, 1986, §IV.D; Fienup and Wackerman, 1986).

To examine the effect of contaminated magnitude data, contamination $C[p,q]$ is here added to $F[p,q]$ (the spectrum of $f[m,n]$). The contaminated magnitude data $M[p,q]$ is then computed as $M[p,q] = |F[p,q] + C[p,q]|$. The pixels of the real and imaginary parts of $C[p,q]$ are pseudo-randomly independently Gaussian distributed with zero mean. It is also ensured that $C[p,q]$ is conjugate symmetric, so that $M[p,q]$ is the Fourier transform magnitude of a real image. It is straightforward to form

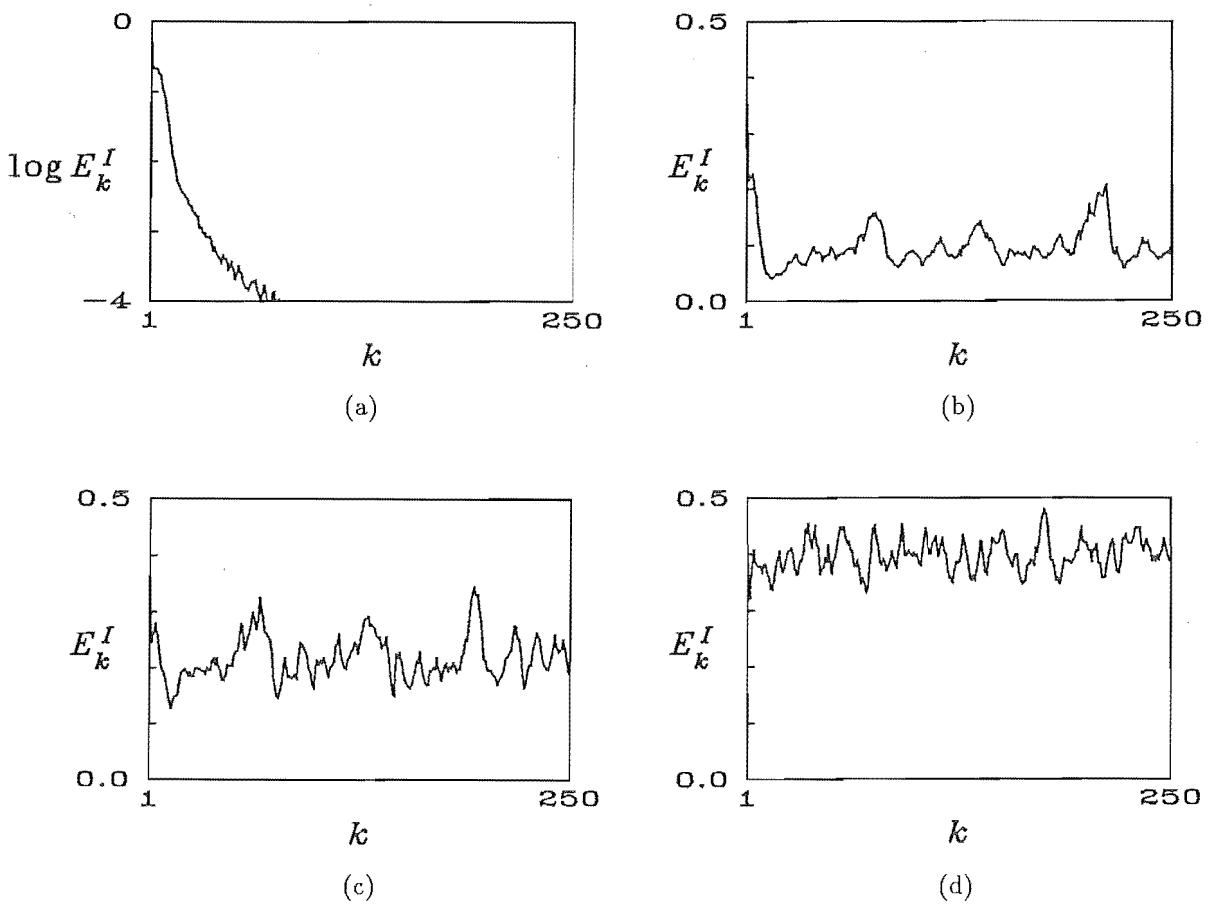


Figure 4.2 Error curves obtained when hybrid input-output algorithm operates on (a) uncontaminated and (b–d) contaminated ($snr = 16, 4, 1$) magnitude of Fourier transform of first original image. Note logarithmic scale of abscissa in (a).

$C[p, q]$ with such characteristics by generating $c[m, n]$, the pixels of which are real and uniformly distributed with zero mean. The *signal to noise ratio*, denoted by snr, is here defined as

$$snr = \mathrm{En}\{F[p, q]\} / \mathrm{En}\{C[p, q]\} \tag{4.3}$$

Figure 4.2b–d shows the error curves obtained when the hybrid input-output algorithm is applied to the contaminated Fourier magnitude of Figure 4.1, for various levels of contamination. It is evident that E_k^{I} oscillates erratically for all three contamination levels shown. Note that the smallest value to which E_k^{I} falls in each case tends to increase as the contamination level is increased. It is worth mentioning that the average range of the oscillations in E_k^{I} (i.e. the mean difference between the E_k^{I} values at each peak and the following trough in the error curve) appears to depend upon the intricacy of the original image. Computational experience indicates that this range tends to decrease as the intricacy of the original image increases.

It is informative to study this erratic behaviour of the hybrid input-output algorithm further. Before doing so so, it is convenient to introduce the *reconstruction*

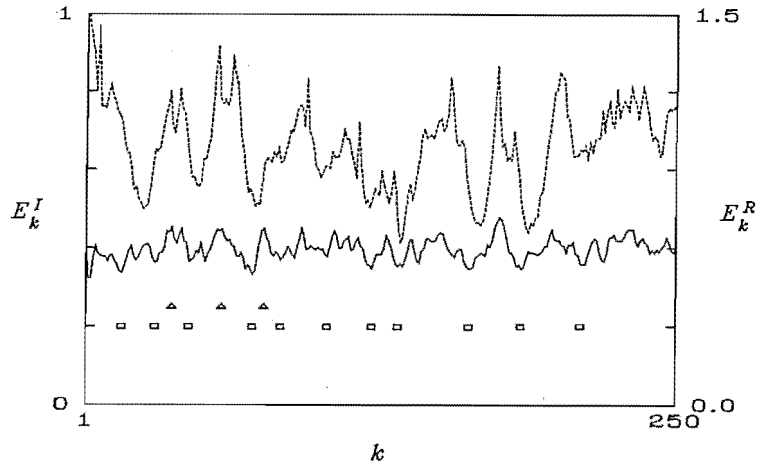


Figure 4.3 Image error E_k^{I} (solid line) and reconstruction error E_k^{R} (dashed line) for $snr=1$, $\beta=\frac{1}{2}$.

error E_k^{R} of $\bar{f}_k[m,n]$, as defined by

$$E_k^{\mathrm{R}} = \mathrm{En}\left\{\mathrm{Reg}\{\bar{f}_k[m,n], f[m,n]\} - f[m,n]\right\} / \mathrm{En}\{f[m,n]\} \tag{4.4}$$

where $\mathrm{Reg}\{\bar{f}_k[m,n], f[m,n]\}$ denotes the *registration* of $\bar{f}_k[m,n]$ against $f[m,n]$. Such registration is necessary because the version of $f[m,n]$ contained in $\bar{f}_k[m,n]$ is likely to be translated and possibly rotated with respect to $f[m,n]$, as explained in §2.1. The registration of $a[m,n]$ against $b[m,n]$ is here defined as

$$\mathrm{Reg}\{a[m,n], b[m,n]\} = a[s\,m+\gamma, s\,n+\delta]\,, \tag{4.5}$$

where $s = \pm 1$ and where s, γ and δ are chosen such that

$$\sum_m \sum_n a[s\,m+\gamma, s\,n+\delta]\, b[m,n] \tag{4.6}$$

is maximized. In practice, s is determined by comparing the height of the largest peak in $a[m,n] * b[m,n]$ with the height of the largest peak in $a[-m,-n] * b[m,n]$. Then γ and δ are determined from the coordinates of the highest of these peaks.

Both the reconstruction error E_k^{R} and the image error E_k^{I}, obtained when the hybrid input-output algorithm operates on the contaminated ($snr=1$) magnitude of the Fourier transform of the image shown in Figure 4.1, are plotted in Figure 4.3. Note that the peaks in E_k^{R} tend to correspond to peaks in E_k^{I}, and that the troughs in E_k^{R} usually correspond to troughs in E_k^{I}. Since E_k^{R} cannot be computed in practical applications of phase retrieval, because $f[m,n]$ is then of course unknown, the aforesaid correspondence between E_k^{R} and E_k^{I} suggests that the more faithful $\bar{f}_k[m,n]$ are those for which k is such that E_k^{I} is locally minimum.

It is informative to examine $\bar{f}_k[m,n]$ for various values of k. Figure 4.4a–i shows the $\bar{f}_k[m,n]$ obtained at the local minima of the error curve marked as $\square$ in Figure 4.3, whereas Figure 4.4j–l shows the $\bar{f}_k[m,n]$ obtained at the local maxima of the error curve marked as $\triangle$. Note that each of the reconstructed images in Figure 4.4 has been registered with the original image to aid comparison. On comparing Figure 4.4a–i and Figure 4.4j–l with Figure 4.1, it is evident that the more faithful $\bar{f}_k[m,n]$ are indeed those corresponding to local minima in the error curve.

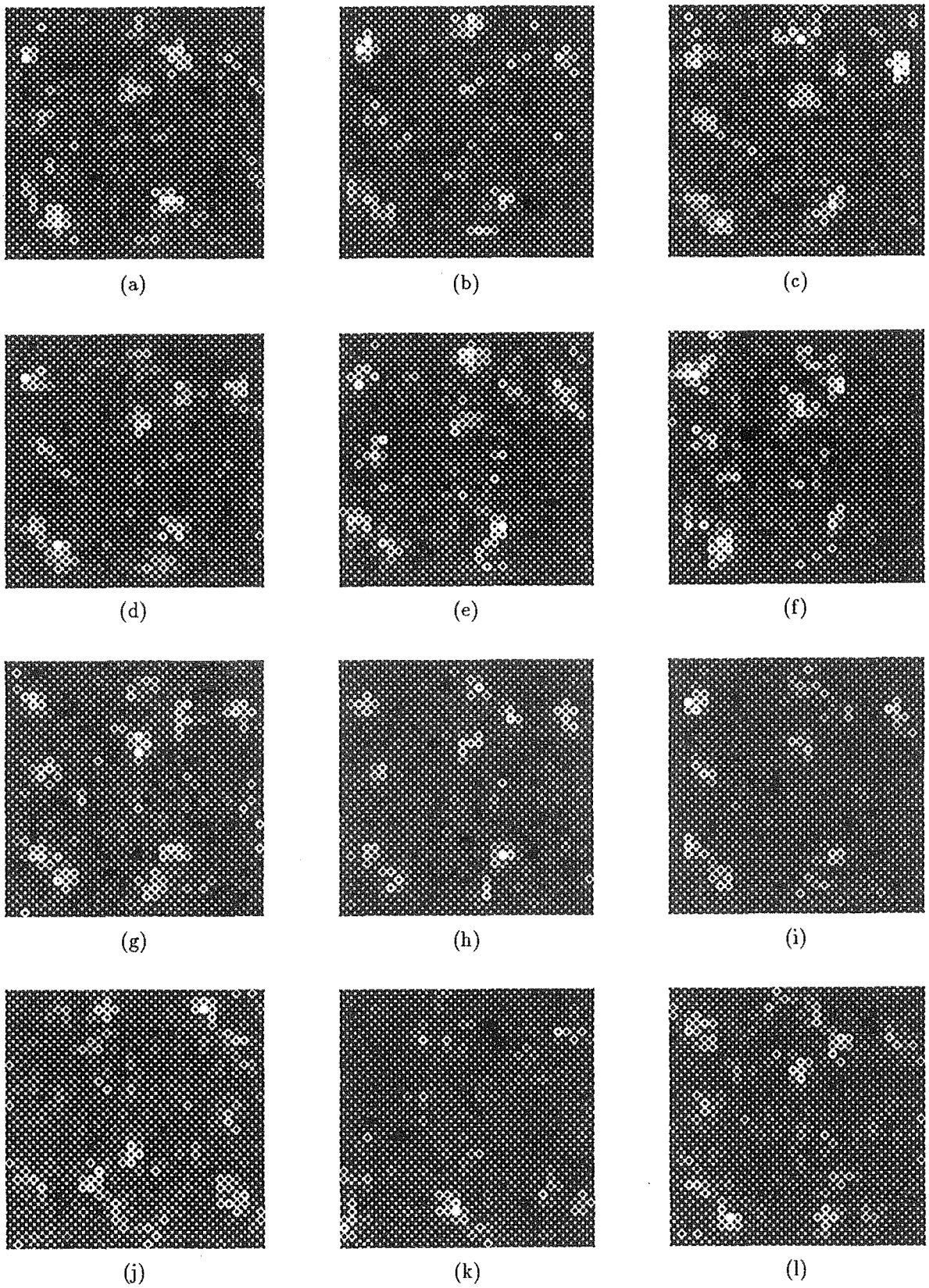


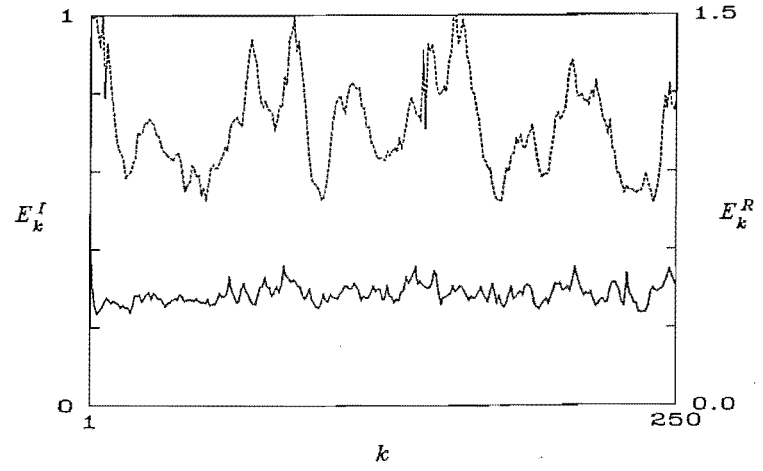
Figure 4.4 Reconstructed images corresponding to (a–i) first 9 points marked □ and (j–l) 3 points marked △ in Figure 4.3.

It is worth noting that those $\bar{f}_k[m,n]$ corresponding to local minima in the error curve for which E_k^{I} is relatively large (e.g. Figure 4.4e,f) tend to be less faithful than those which correspond to local minima for which E_k^{I} is relatively small (e.g. Figure 4.4h). However, reconstructions from local minima with nearly identical values of E_k^{I} can often differ significantly in their faithfulness (e.g. Figure 4.4a,h). These examples serve to illustrate the tenuous connection between the faithfulness of a particular $\bar{f}_k[m,n]$ and the value of E_k^{I} corresponding to it.

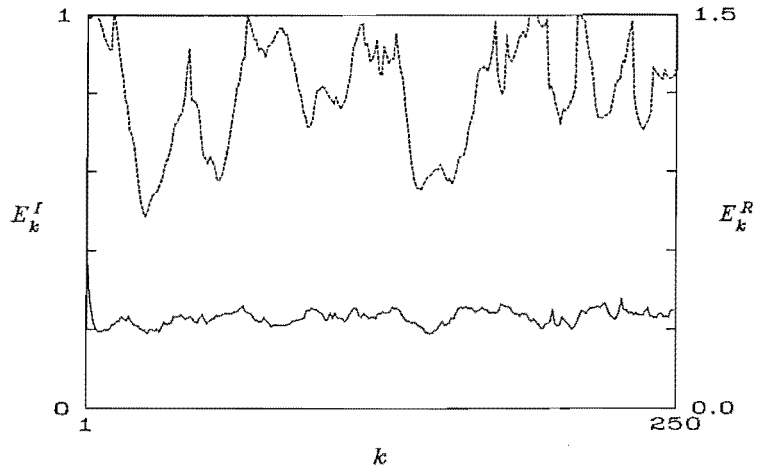
The reason for the erratic behaviour of the hybrid input-output algorithm, when it operates on contaminated data, is now explained. As described in §3.2.8.2, the input $\tilde{f}[m',n']$, at a particular pixel $[m',n']$ for which the output pixel $\bar{f}[m',n']$ violates the image space constraints, is increased (decreased) repeatedly until the output pixel $\bar{f}[m',n']$ becomes positive (negative). But when the magnitude data are contaminated, the image space constraints can never be satisfied exactly. Once a "reasonable" solution has been found, the algorithm continues to increase (or decrease) the values of some pixels of the input in an attempt to exactly satisfy the image space constraints. In so doing, such pixels can assume significantly erroneous values, which may actually drive the output image further away from the true solution.

The parameter β appearing in (3.59) is similar to a gain. It is thus reasonable to question the effect that reducing the value of β has upon the behaviour of the hybrid input-output algorithm, when operating on noisy data. On comparing Figures 4.3, 4.5a, and 4.5b, it is evident that the algorithm also behaves erratically with small values of β, but that the rate of oscillation of E_k^{I} tends to decrease as β is reduced. In addition, the minimum value to which E_k^{I} falls tends to decrease as β is reduced. Note, however, that the minimum value to which E_k^{R} falls is roughly similar for all three values of β. The faithfulness of the reconstructions corresponding to the smallest minima in each of Figures 4.5a and 4.5b is similar to that of the reconstructions displayed in Figure 4.4a–i. This illustrates the point that the value of E_k^{I} should be used with caution when attempting to assess the comparative faithfulness of reconstructed images. Thus the adoption of a smaller value for β, whilst reducing the rate of oscillation of E_k^{I}, need not significantly improve the faithfulness of the resulting reconstructed images.

Because of the erratic behaviour of the hybrid input-output algorithm, simple strategies are unsatisfactory when the data are appreciably contaminated. For example, the common strategy of performing one or more cycles comprising a number of hybrid input-output iterations followed by a number of error-reduction iterations is often ineffective. This is especially so if E_k^{I} happens to be comparatively large after the hybrid input-output iterations in a particular cycle are finished, because little subsequent improvement to the reconstructed image is made by the error-reduction iterations. Useful strategies must either invoke a modified hybrid input-output algorithm with improved behaviour, if such can be devised, or appropriately exploit the standard hybrid input-output algorithm. An example of the former approach is presented in §4.3, whereas §4.4 presents an example of the latter approach.



(a)



(b)

Figure 4.5 Image error E_k^{I} (solid line) and reconstruction error E_k^{R} (dashed line) for $snr = 1$ with (a) $\beta = \frac{1}{4}$ and (b) $\beta = \frac{1}{8}$.

4.3 Incorporating a Threshold into the Hybrid Input-Output Algorithm

It has been explained in §4.2 that the erratic behaviour of the hybrid input-output algorithm, when operating on contaminated data, is due to the algorithm attempting to exactly satisfy the image space constraints. This suggests that a modified algorithm, which only attempts to satisfy the image space constraints approximately, may be better behaved. One such possible modification of the hybrid input-output algorithm is the incorporation of a threshold. This involves accepting (i.e. considering that the image space constraints have been satisfied at) those pixels of the output $\bar{f}_k[m,n]$ which have absolute value less than some threshold and that are either outside the image mask or have negative value within the image mask.

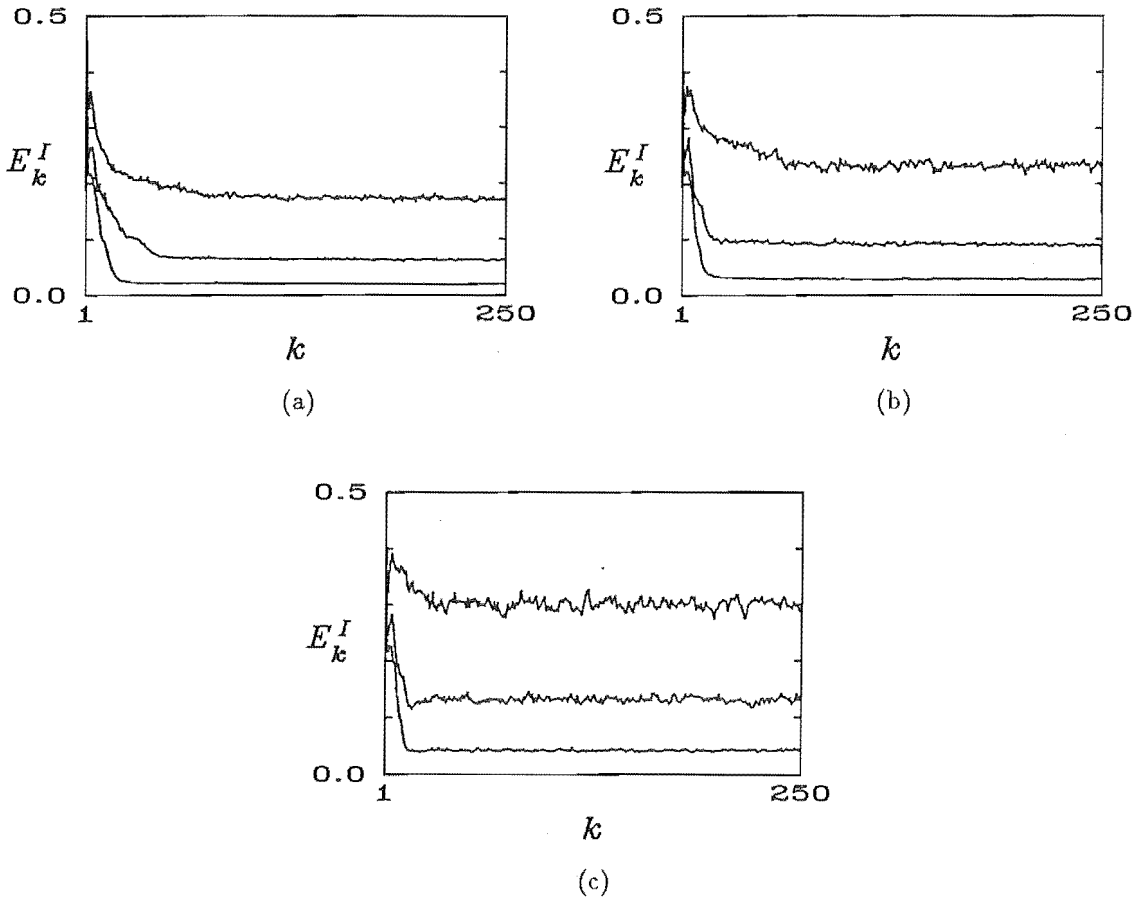


Figure 4.6 Error curves obtained for $snr = 1$ (upper curves), $snr = 4$ (middle curves), and $snr = 16$ (lower curves), with threshold level of (a) $tl = \max\{c[m,n]\}/4$, (b) $tl = \max\{c[m,n]\}/8$, (c) $tl = \max\{c[m,n]\}/16$.

A threshold can be conveniently incorporated into the hybrid input-output algorithm by merely redefining V_k (the set of pixels at which the image space constraints are violated, as introduced in §3.2.8.1). V_k is now computed as

$$\begin{array}{rlll} [m,n] \in V_k & \text{if} & |\bar{f}_k[m,n]| > tl & \text{and} \quad mask[m,n] = 0 \\ & \text{or} & \bar{f}_k[m,n] < -tl & \text{and} \quad mask[m,n] = 1 \end{array} \tag{4.7}$$

where tl is the *threshold level*.

Figure 4.6 shows the error curves obtained when the hybrid input-output algorithm, with a threshold incorporated, operates on the contaminated magnitude of Figure 4.1, for various values of tl and snr. For each case, the $\bar{f}_k[m,n]$ for which E_k^{I} is minimum is shown in Figure 4.7. Since, in these examples, the contamination $c[m,n]$ is known, it is convenient to specify tl in terms of the maximum value of $c[m,n]$. The notation $\max\{\cdot\}$ is here invoked to denote the maximum value of an image.

Comparison of the error curves in Figure 4.6 with those in Figure 4.2b–d reveals that the behaviour of the algorithm is certainly improved by the incorporation of a threshold. Note that the range of the oscillations in E_k^{I} tends to increase

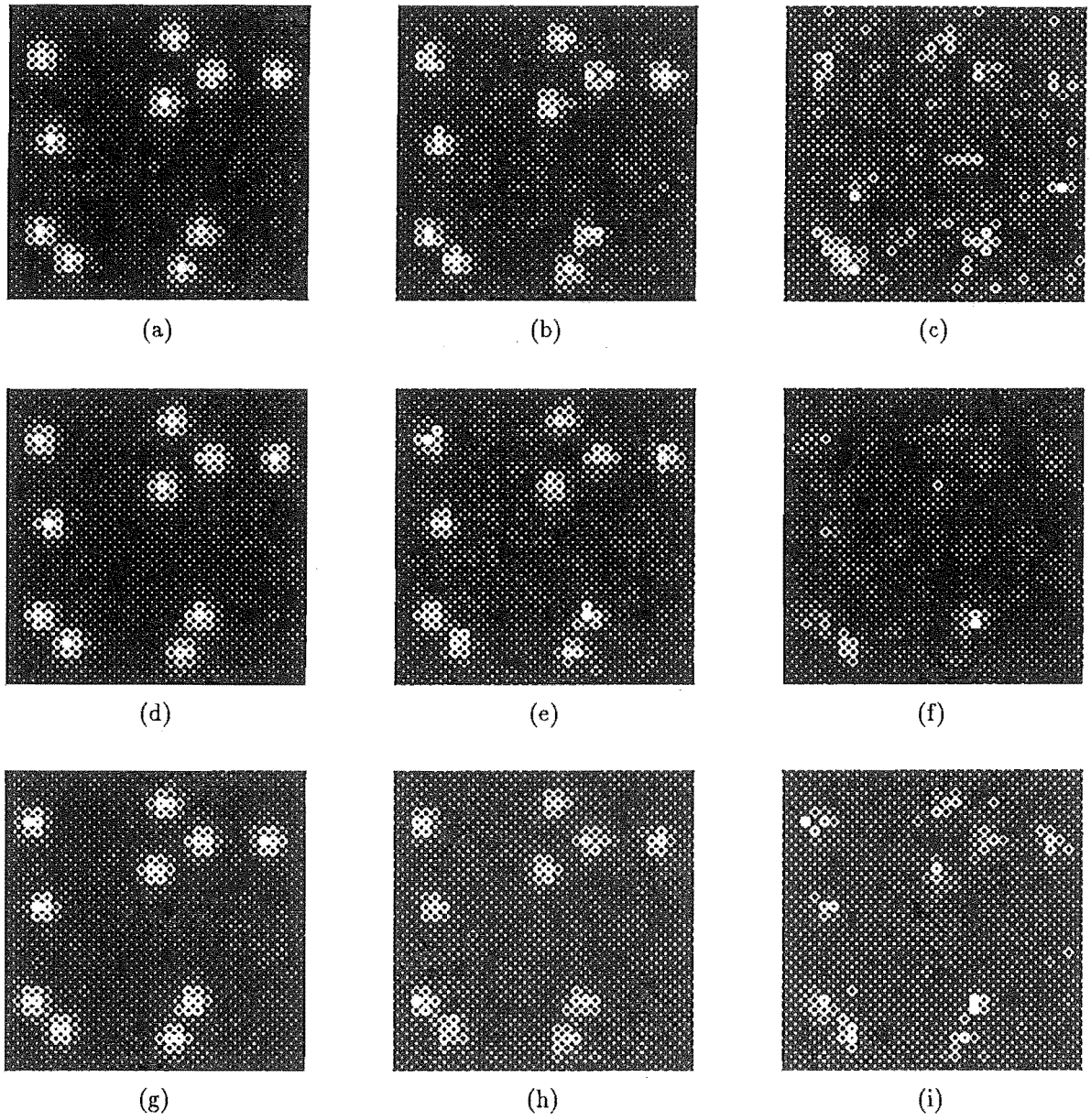


Figure 4.7 Reconstructed images, for $snr = 16, 4, 1$, with threshold level of (a–c) $tl = \max\{c[m,n]\}/4$, (d–f) $tl = \max\{c[m,n]\}/8$, (g–i) $tl = \max\{c[m,n]\}/16$.

as the value of tl is decreased. However, it is necessary to examine the corresponding reconstructed images, to allow the effectiveness of the threshold algorithm to be judged.

Examination of Figure 4.7a,d,g suggests that the use of a threshold produces acceptable reconstructions when the contamination is slight ($snr = 16$). Inspection of Figure 4.7c,f,i reveals that the most faithful reconstruction when the contamination is severe ($snr = 1$) results from the smallest of the three threshold levels that are invoked. However, comparison of the reconstructed image presented in Figure 4.7i (which is computed with $tl = \max\{c[m,n]\}/16$) with those shown in Figure 4.4a–i (which are computed with $tl = 0$) indicates that the best reconstruction that is obtained when a

threshold is employed is not significantly more faithful than the best reconstruction that is obtained without a threshold. Computational experience suggests that this is usually so when the contamination is severe.

It is thus apparent that the incorporation of a threshold into the hybrid input-output algorithm, whilst possibly being useful when the magnitude data are slightly contaminated, is of little benefit when the data are appreciably contaminated. A strategy is detailed in §4.4 which exploits, rather than attempting to eradicate, the erratic behaviour of the hybrid input-output algorithm (as described in §4.2).

4.4 Averaging of Minimal Image-Forms

It has been demonstrated in §4.2 that, when applying the hybrid input-output algorithm to appreciably contaminated magnitude data, the more faithful reconstructions tend to be those which correspond to local minima in the error curve. The outputs $\bar{f}_k[m,n]$ at values of k for which E_k^{I} is a local minimum are here called *minimal image-forms*. Recall from §4.2 that the relationship between the faithfulness of any such minimal image-form and its corresponding value of E_k^{I} is somewhat tenuous. In other words, the minimal image-form possessing the smallest E_k^{I} is not necessarily the most faithful minimal image-form. This suggests that, rather than attempting to select the single "best" minimal image-form, an improved image-form might be generated by appropriately averaging a number of minimal image-forms. Furthermore, by considering each minimal image-form as a contaminated version of the original image, it is evident that the average of a number of minimal image-forms could well be a more faithful estimate of the true image than even the "best" minimal image-form.

The error curve often possesses so many minima that it can be inefficient to average all the minimal image-forms. There is also a tendency for minimal image-forms corresponding to comparatively shallow minima of the error curve to be significantly less faithful than those corresponding to more pronounced minima. Consequently, "candidate" image-forms must be selected from the set of available minimal image-forms.

It is convenient to denote the value of k corresponding to the λ^{th} minimal image-form by k_λ. The λ^{th} minimal image-form is selected as a *candidate image-form* if $E_{k_\lambda}^{\mathrm{I}}$ is the minimum value of E_k^{I} throughout the range of k spanning μ values on either side of k_λ. Thus $\bar{f}_{k_\lambda}[m,n]$ is a candidate image-form if

$$E_{k_\lambda}^{\mathrm{I}} = \min_{k_\lambda - \mu \leq k \leq k_\lambda + \mu} E_k^{\mathrm{I}} \tag{4.8}$$

The j^{th} candidate image-form, of which there are J in total, is denoted by $cand_j[m,n]$.

Because the version of the original image contained in each candidate image-form may be differently located and orientated, it is necessary to register (in the sense described in §4.2) each candidate image-form before averaging the candidates. The proposed registration and averaging procedure is outlined below in Algorithm 4.1. The sum of the first j registered candidate image-forms is denoted by $sum_j[m,n]$. The average of the J registered candidate image-forms is here termed the *enhanced image-form*, and is denoted by $e[m,n]$.

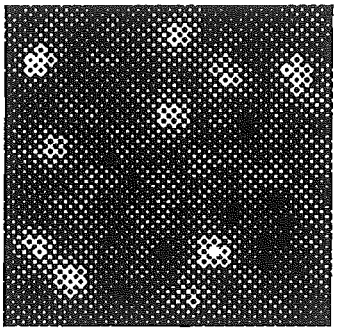


Figure 4.8 Enhanced image-form computed by averaging strategy ($snr = 1$, $\mu = 10$).

Algorithm 4.1

Step 1: Set $sum_1[m,n] = cand_1[m,n]$.

Step 2: Repeat the following step, for $j = 2$ to J:

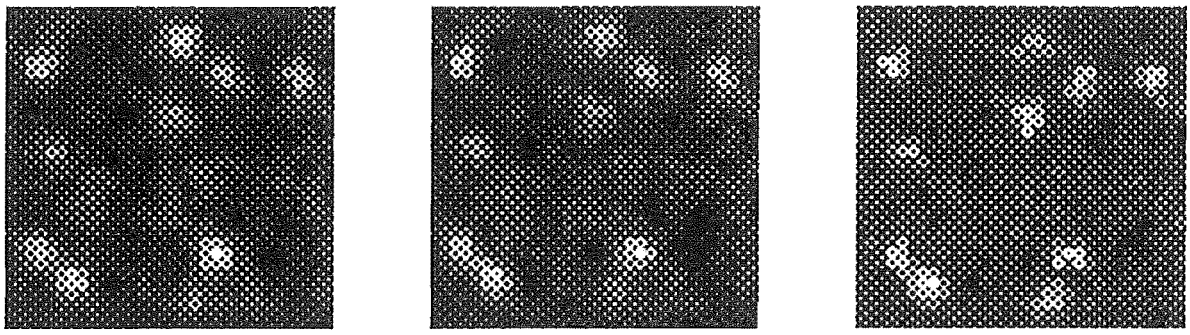
Step 2.1: Compute $sum_j[m,n] = sum_{j-1}[m,n] + \mathrm{Reg}\{cand_j[m,n], sum_{j-1}[m,n]\}$.

Step 3: Compute $e[m,n] = \frac{1}{J} sum_J[m,n]$.

Figure 4.8 shows $e[m,n]$ as computed by Algorithm 4.1 (with $\mu = 10$) from the contaminated ($snr = 1$) magnitude of the Fourier transform of the image shown in Figure 4.1. The 11 candidate image-forms selected automatically by the method described in the paragraph containing (4.8) correspond to the values of k marked $\square$ on Figure 4.3. It is evident that the enhanced image-form shown in Figure 4.8 is a significant improvement over each of the candidate image-forms shown in Figure 4.4a–i.

It is worthwhile examining the effect that the parameter μ in (4.8) has upon the faithfulness of the enhanced image-form. Figure 4.9 shows $e[m,n]$ computed from the contaminated ($snr = 1$) magnitude of Figure 4.1, with $\mu = 1, 3, 30$, for which the corresponding values of J are $J = 50, 28, 3$. These enhanced image-forms should be compared with Figure 4.8, for which $\mu = 10$ and $J = 11$. It is evident that as the value of μ is increased, the corresponding value of J tends to decrease, and consequently $e[m,n]$ becomes noisier. Although this suggests that a small value of μ is to be preferred, it seems that very small values of μ should be avoided, for the following reason. When $\mu = 1$, it is clear that every minimal image-form is selected as a candidate image-form. Thus minimal image-forms from shallow minima in the error curve, which tend to be significantly less faithful than those from pronounced minima, are incorporated into $e[m,n]$. In some cases, this may well reduce the faithfulness of $e[m,n]$. Some of the factors that should be considered when choosing a value for μ are mentioned in §4.6.

A number of further results of the averaging strategy described in this section are presented in §4.5.



(a) (b) (c)

Figure 4.9 Enhanced image-forms computed by averaging strategy ($snr = 1$) for (a) $\mu = 1$, (b) $\mu = 3$, (c) $\mu = 30$.

4.5 Further Illustration of the Averaging Strategy

Although the results of the averaging strategy presented in §4.4 are encouraging, they are only for a single original image, and might not therefore be representative of general image-forms reconstructed from contaminated Fourier magnitudes. Further results are now presented for various specimen images and various noise levels. It is convenient to refer to the candidate image-form for which E_k^I is minimum as the *master candidate*. To allow the assessment of the effectiveness of the averaging strategy, both the master candidate (which tends to be the best of the candidate image-forms) and the enhanced image-form are shown for each case.

Figure 4.10 shows a second original image, comprising the same number of isolated "blobs" as in Figure 4.1, and master candidates and enhanced image-forms for various noise levels. Note the marked improvement in each case in the faithfulness of the shape and brightness of the blobs in the enhanced image-form over the master candidate (especially at high noise levels), and the lower level of background noise. Figure 4.11 shows an original image consisting of blobs of varying brightness, together with several reconstructed images. Figure 4.12 shows corresponding results for an original image comprising blobs of varying diameter. Note that at high noise levels, the smaller and dimmer blobs are poorly reconstructed, since they cause smaller fringes on the Fourier magnitude than do the larger blobs, and hence are affected to a greater extent by the noise.

Figure 4.13 confirms that the averaging strategy can be usefully applied to images exhibiting quite intricate detail. It should be mentioned that the faithfulness of the candidate image-forms, which are averaged to form each of the enhanced image-forms shown in Figure 4.13e–g, is highly variable. Improved schemes for selecting candidate image-forms are suggested in §4.6.

Let it be assumed that the candidate image-forms only differ from one another in their high spatial frequency components. The enhanced image-form resulting from the averaging strategy would then merely be a low-pass filtered version of any one of the candidate image-forms. In this case, an image-form obtained by applying the standard hybrid input-output algorithm to a similarly low-pass filtered version of

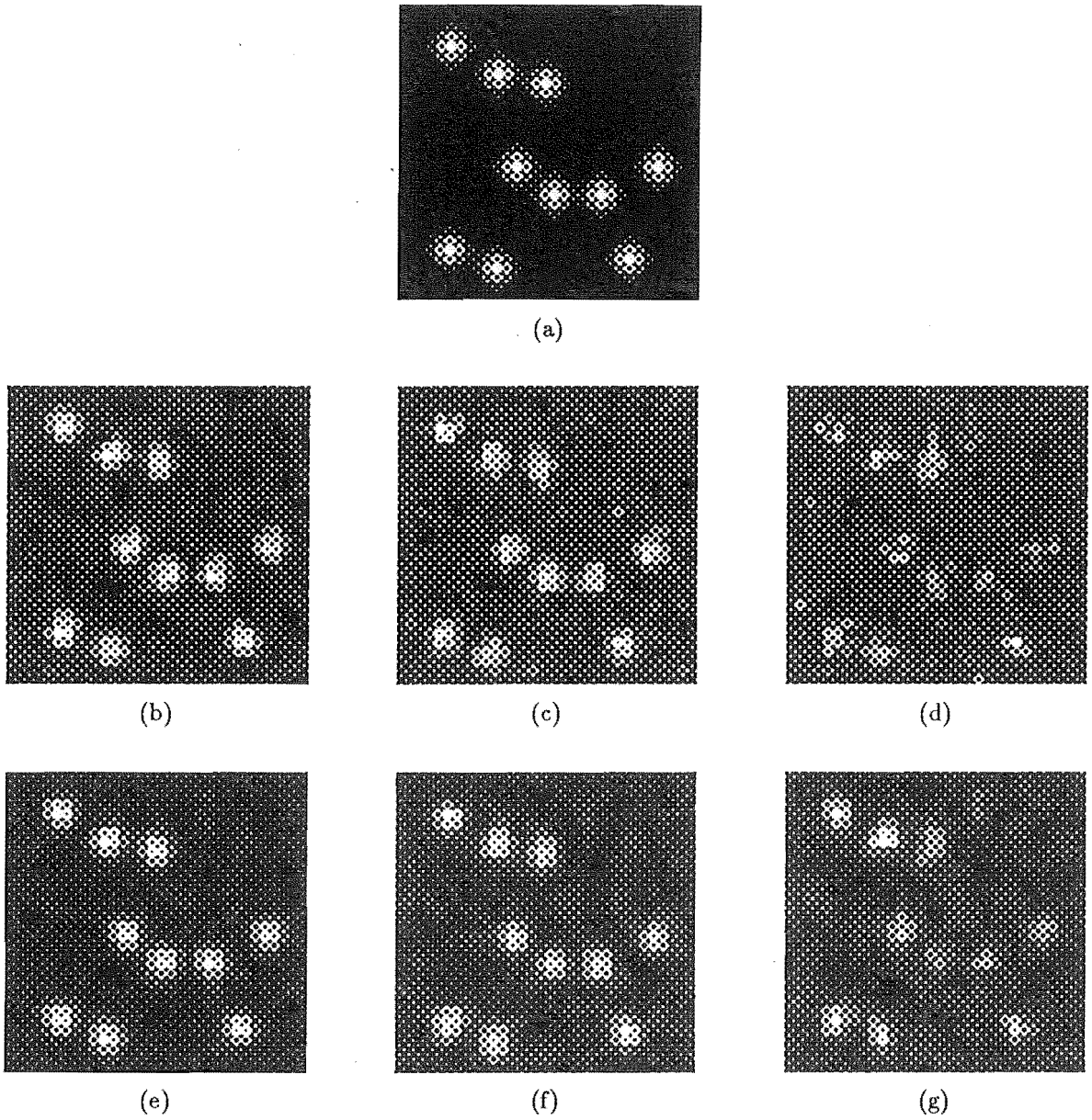


Figure 4.10 Application of averaging strategy ($\mu = 10$) to magnitude of Fourier transform of second original image. (a) original image, (b–d) master candidates, and (e–g) enhanced image-forms for $snr = 16, 4, 1$.

the Fourier magnitude data would then be similar to the enhanced image-form. This would then imply that the averaging strategy is merely a computationally expensive means by which to recover an image-form from low-pass filtered Fourier magnitude data. However, inspection of Figure 4.4a–i reveals that the assumption at the beginning of this paragraph is not valid. Although there is a significant difference between the high spatial frequency components of the candidate image-forms in Figure 4.4a–i, there is also a significant difference between their the low spatial frequency components, as is evidenced by the absence of some of the blobs in a number of the candidate image-forms.

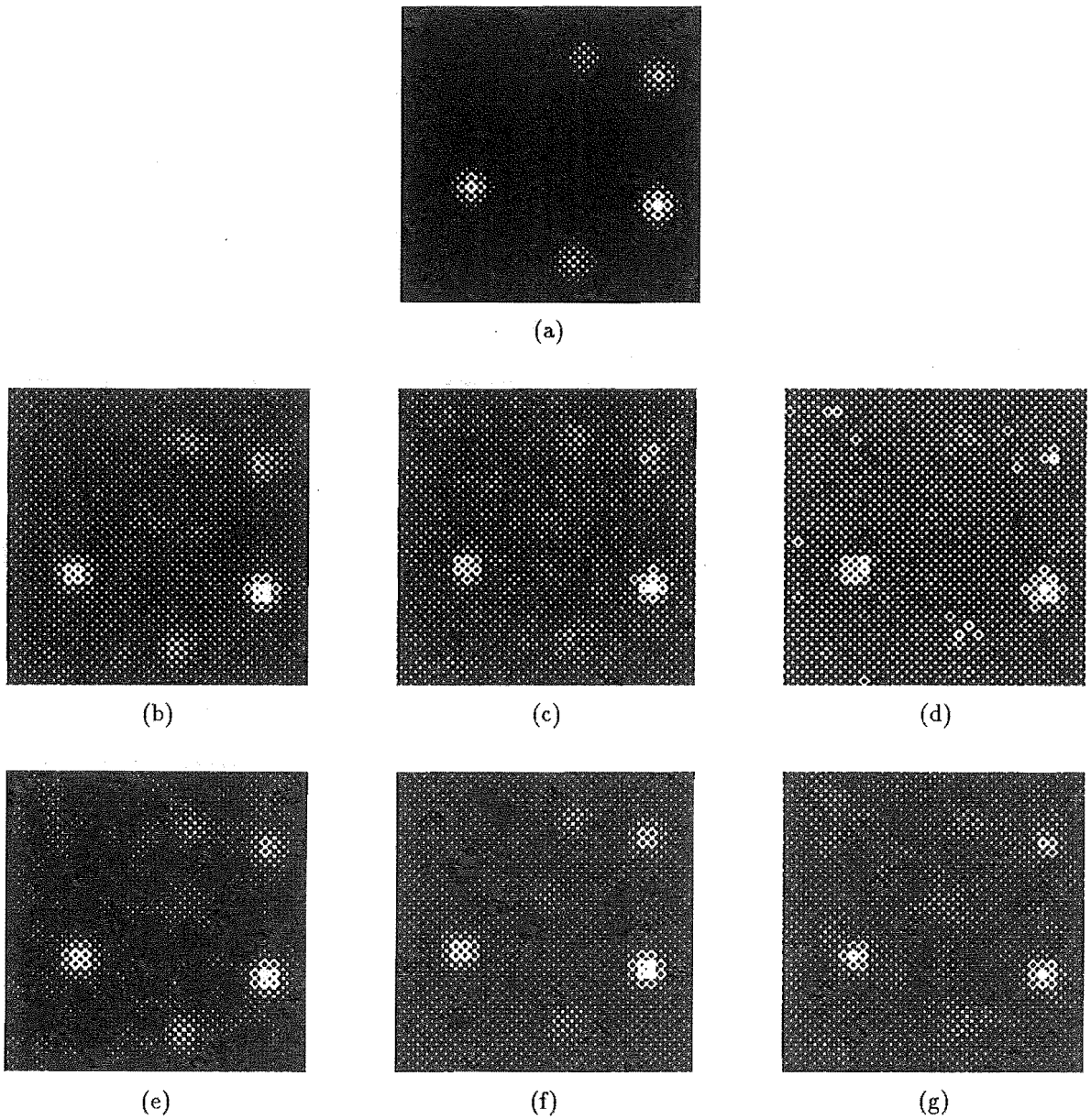


Figure 4.11 Application of averaging strategy ($\mu = 10$) to magnitude of Fourier transform of third original image. (a) original image, (b–d) master candidates, and (e–g) enhanced image-forms for $snr = 16, 4, 1$.

It is now worthwhile to show by example that the averaging strategy can operate effectively in situations where it is undesirable to low-pass filter (or, equivalently, window) the Fourier magnitude data. Figure 4.14 shows an original image comprising a number of isolated pixels, all of the same brightness. Note particularly the two bright pixels separated from each other by a single dark pixel. The pixels of the master candidate (Figure 4.14b) corresponding to the bright pixels of the original image show pronounced variation in intensity, the dimmest one being little brighter than the spurious detail. The enhanced image-form (Figure 4.14c), however, is a significant improvement, since the variation in intensity of the bright pixels is con-

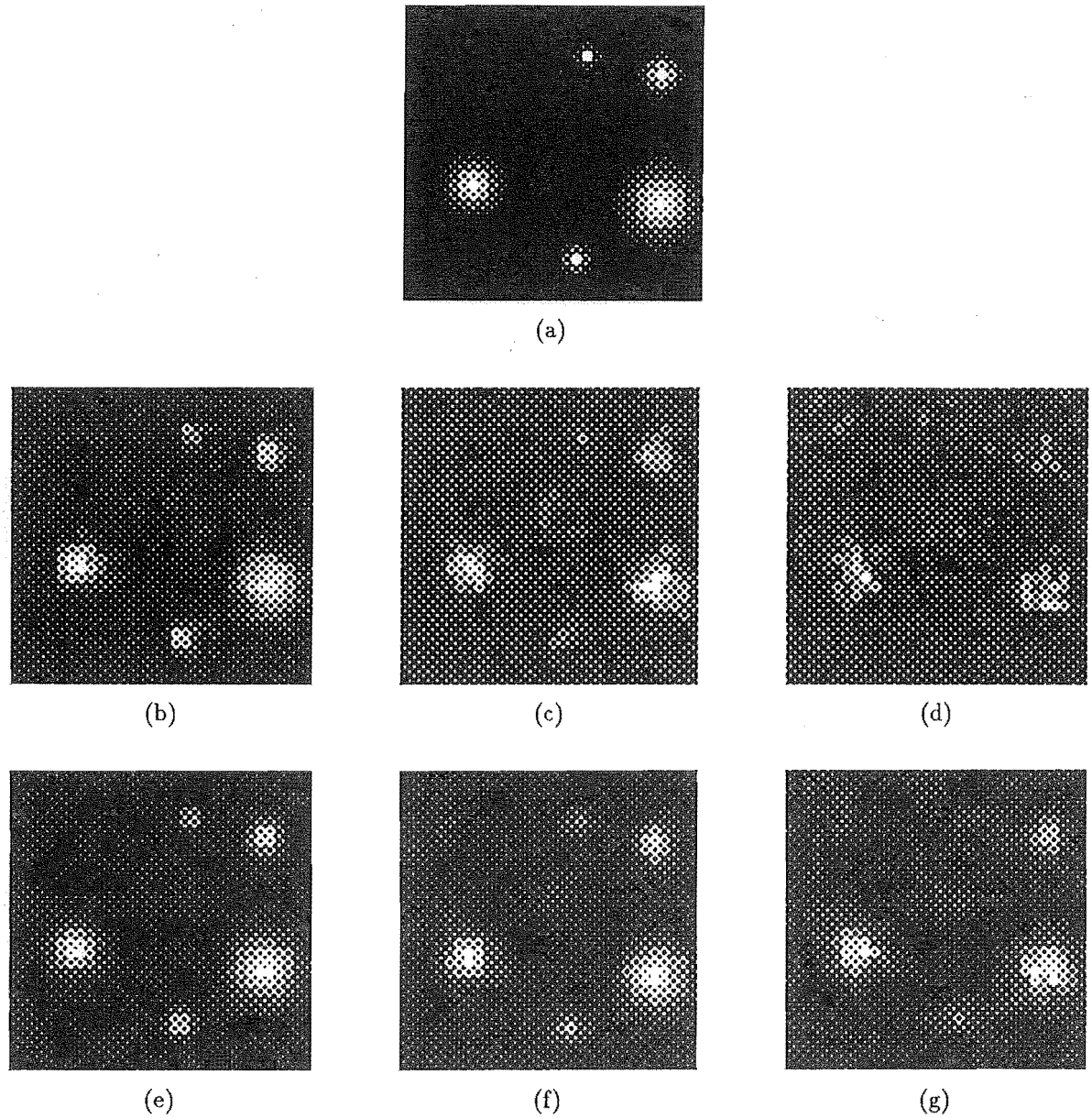
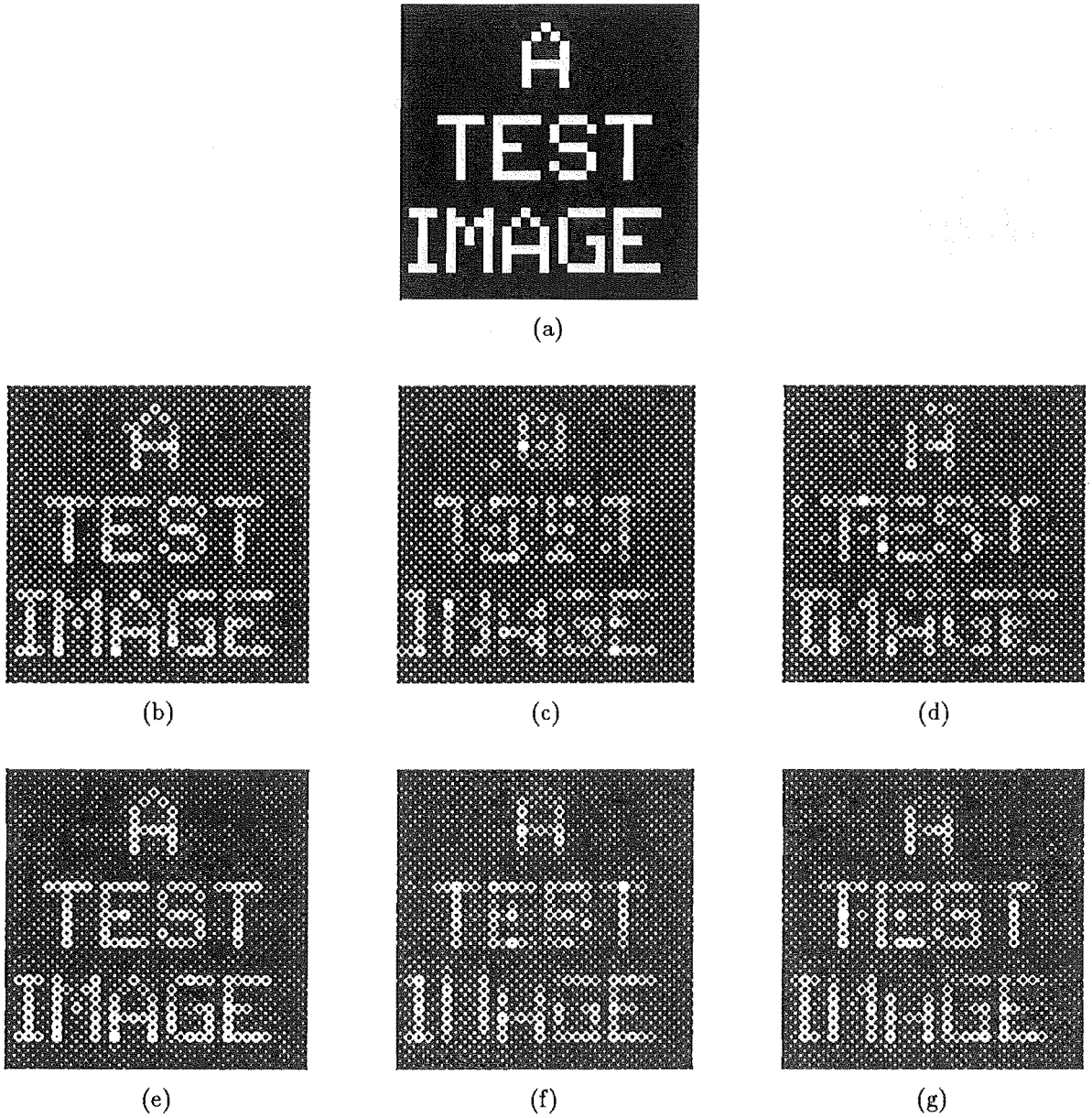


Figure 4.12 Application of averaging strategy ($\mu = 10$) to magnitude of Fourier transform of fourth original image. (a) original image, (b–d) master candidates, and (e–g) enhanced image-forms for $snr = 16, 4, 1$.

siderably reduced, and the spurious detail is diminished. Note that the two closely spaced bright pixels are well resolved, which would have been impossible if the given Fourier magnitude was appreciably windowed.

4.6 Discussion

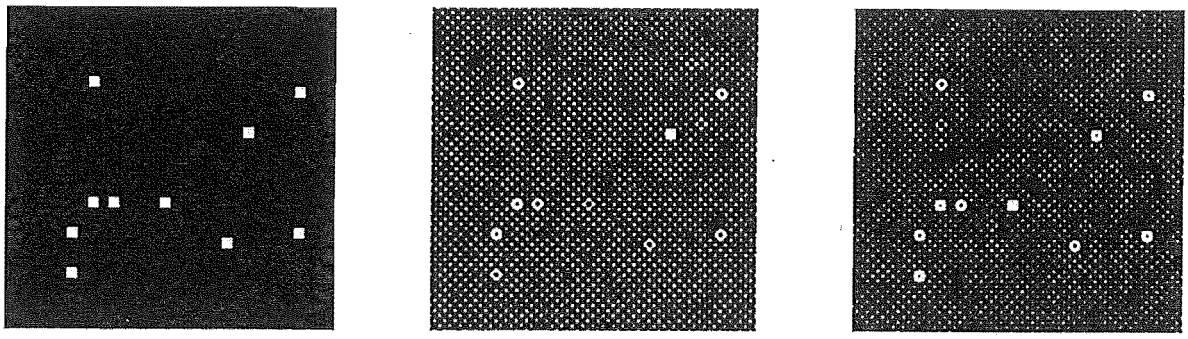
It has been demonstrated in §4.2 that the hybrid input-output algorithm, when operating on significantly contaminated magnitude data, gives rise to pronounced fluctuations in the image error. The incorporation of a threshold into the hybrid input-



(a) (b) (c) (d) (e) (f) (g)

Figure 4.13 Application of averaging strategy ($\mu = 20$) to magnitude of Fourier transform of fifth original image. (a) original image, (b–d) master candidates, and (e–g) enhanced image-forms for $snr = 16, 12, 8$.

output algorithm, as described in §4.3, appears to be of little advantage. However, it it has been shown that the averaging strategy introduced in §4.4, which is illustrated in §4.5, is effective for generating useful reconstructions from significantly contaminated magnitude data. This averaging strategy is straightforward to implement, but is somewhat computationally intensive due to the number of iterations required to generate a sufficient number of candidate image-forms. Numerous modifications to this strategy are possible, some of which are now described. Improvements to the averaging strategy may involve either enhancing the method by which the candidate image-forms are generated, or enhancing the registration procedure.



(a) (b) (c)

Figure 4.14 Application of averaging strategy ($\mu = 10$) to magnitude of Fourier transform of sixth original image. (a) original image, (b) master candidate, and (c) enhanced image-form for $snr = 1.5$.

The examples presented in Figure 4.9 indicate that the choice of μ in the averaging strategy described in §4.4 is an important consideration. Computational experience suggests that the optimum value of μ varies with the total number of iterations performed, the contamination level of the data, and the intricacy of the original image. It would be desirable to determine automatic strategies by which to choose μ.

To avoid the need to choose a value for μ, methods more sophisticated than that described in §4.4 for selecting the candidate image-forms from the set of minimal image-forms could well be devised. For example, one could select as candidate image-forms only those minimal image-forms which have a "significant" correlation with a number of other minimal image-forms. Alternatively, it may prove advantageous to perform some further processing on each candidate image-form before averaging is performed. However, experiments indicate that performing a number of iterations of error-reduction (or error-reduction with a threshold incorporated as described in §4.3) on each candidate image-form does not significantly improve the faithfulness of the resultant enhanced image-form.

The registration procedure described in §4.4 could well be improved. Consideration of Algorithm 4.1 reveals that the manner in which the j^{th} candidate image-form is registered depends on the faithfulness of $sum_{j-1}[m,n]$, which tends to improve as j increases. It may therefore be advantageous to calculate $e[m,n]$ as previously described, then register each of the J candidate image-forms against $e[m,n]$, and finally average these registered candidate image-forms to produce a better enhanced image-form.

It is worth mentioning the difficulty encountered in attempting to determine the relative merits of various phase retrieval strategies. To make a meaningful comparison of two strategies, it is essential to compare the reconstructions of a significant number of images (and preferably of a number of different image types) for various contamination levels. The point being made here is that attempting to compare two strategies by simply comparing the reconstructions of a single specimen image can often lead to erroneous conclusions.

Although the averaging strategy is an effective means of exploiting the erratic

behaviour of the hybrid input-output algorithm, it is still worthwhile seeking algorithms which are better behaved in the presence of severe contamination on the data. It should be pointed out that the hybrid input-output algorithm enforces the Fourier space constraint exactly at each iteration, while urging the reconstructed image to satisfy the image space constraints. However, the true image satisfies (by definition) the image space constraints exactly, but does not satisfy the Fourier constraint exactly when the magnitude data are contaminated. This suggests that an iterative algorithm which operates by enforcing the image space constraints exactly at each iteration, while urging the reconstructed image to satisfy the Fourier space constraint, may be more effective than the hybrid input-output algorithm when the data are significantly contaminated. It is worth noting that the simulated annealing algorithm for phase retrieval (see §3.2.9.2) is of this type. The possibility of modifying the hybrid input-output algorithm so that it operates in the above manner is discussed further in §8.2.1.

Chapter 5

Reducing the Computational Requirements of Phase Retrieval Techniques

All currently available phase retrieval algorithms are computationally intensive. In this chapter, various new techniques are presented for reducing the computational effort that must be expended to effect phase retrieval.

Firstly, the need for such techniques is discussed in §5.1. A new method for accelerating the convergence of the hybrid input-output algorithm, by initially recovering the phases of only the low spatial frequency components of the reconstructed image, is described in §5.2. This method is shown to significantly accelerate the convergence of the hybrid input-output algorithm when reconstructing real-valued images. A new input-output algorithm is described in §5.3, which is shown to be a significant improvement over the hybrid input-output algorithm in situations where only a positivity constraint may be employed in image space. Finally, a new simulated annealing algorithm for phase retrieval, which operates in Fourier space, is described in §5.3.

5.1 Motivation

A number of different phase retrieval techniques are reviewed in §3.2. Of all these techniques, however, there are only two algorithms which have been demonstrated as being capable of converging to an acceptable solution from a random starting image, when operating on contaminated Fourier magnitude data. These are Fienup's (1978) hybrid input-output algorithm (see §3.2.8.2) and the simulated annealing algorithm of Nieto-Vesperinas *et al.* (1988) (see §3.2.9.2). Unfortunately, both methods are computationally intensive, because of their iterative nature. It is thus worthwhile to seek techniques to reduce the computational cost of these algorithms. The computational requirements of the hybrid input-output algorithm and the simulated annealing algorithm are now discussed further.

The major computational expense of the hybrid input-output algorithm is incurred in performing the two FFTs which are required at each iteration. If the available Fourier magnitude data comprises $N \times N$ samples, then each FFT (see §1.6)

requires approximately $(N \log_2 N)^2$ floating point operations. In addition to depending upon N, the total amount of computation required to recover any particular image depends on the number of iterations that are required to reduce the image error E_k^{I} to an acceptable value. The number of iterations required tends to increase with the intricacy of the image being reconstructed. A complex image tends to require a significantly greater number of iterations than a positive image of the same dimensions. A large number of iterations are required if the hybrid input-output algorithm stagnates, unless the procedures described by Fienup and Wackerman (1986) are employed to extricate the algorithm from stagnation.

When recovering images of reasonable size (e.g. 128×128 pixels), especially complex ones, with the hybrid input-output algorithm, CPU times of the order of hours are not uncommon. It is thus evident that any computational saving is highly desirable. A technique is described in §5.2 for accelerating the convergence of the hybrid input-output algorithm, by initially recovering the phases of only a fraction of the spatial frequency components of the reconstructed image.

Some situations exist in which the hybrid input-output algorithm does not always perform satisfactorily. Such a situation can arise when attempting to recover an image from its Fourier magnitude, if positivity is the only constraint that may be applied in image space. Furthermore, the hybrid input-output algorithm appears to have been derived in a rather *ad hoc* manner (Fienup, 1982). It is thus worthwhile to seek alternatives to the hybrid input-output algorithm. One such alternative, which is a member of the class of input-output algorithms (§3.2.8.2), is described in §5.3.

The major computational load of the simulated annealing algorithm for phase retrieval (§3.2.9.2) results from the calculation of the change in the cost function $Q'\{\hat{f}[m,n]\}$ (as defined by (3.69)) following each perturbation. If the reconstructed image comprises $M \times M$ nonzero pixels, then this calculation requires $o(M^2)$ operations. Since each pixel of the reconstructed image is perturbed in one image scan, the computational requirements are $o(M^4)$ operations per image scan. The number of image scans required to adequately recover any particular image depends on many factors, including the intricacy of the image and the choice of the annealing schedule. For an image of reasonable size, thousands of image scans may be required. It is thus worthwhile to seek improvements to the simulated annealing algorithm of Nieto-Vesperinas *et al.* (1988), which may reduce the computational requirements (especially when M is large). An alternative simulated annealing implementation, in which the computational requirements are reduced to $o(M^3)$ operations per scan by operating in Fourier space, is described in §5.4.

5.2 Accelerating the Convergence of the Hybrid Input-Output Algorithm

A new technique for accelerating the convergence of the hybrid input-output algorithm, by initially recovering the phases of only the low spatial frequency components of the reconstructed image, is presented in this section. This method has been described briefly by McCallum and Bates (1989, §6).

The performance of the standard hybrid input-output algorithm, when em-

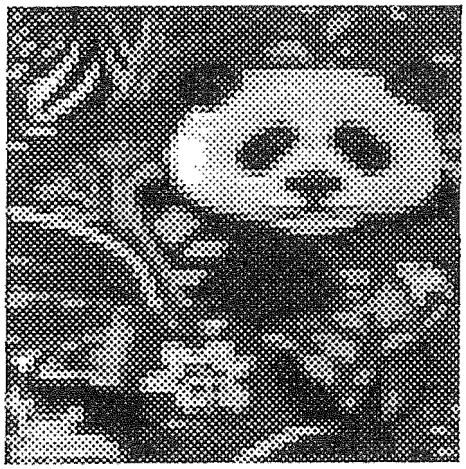


Figure 5.1 Central 64×64 pixel region of original 128×128 pixel positive image.

ployed to recover the phase of intricate images, is described in §5.2.1. The new accelerated hybrid input-output algorithm is detailed in §5.2.2, and computational examples of this algorithm are presented in §5.2.3. Further aspects of the accelerated hybrid input-output algorithm are discussed in §5.2.4.

5.2.1 Performance of the Standard Hybrid Input-Output Algorithm

It is now appropriate to demonstrate the typical performance of the hybrid input-output algorithm (§3.2.8.2) when it operates on the Fourier magnitude of an intricate image. The performance of the accelerated hybrid input-output algorithm, described in §5.2.2, can then be compared with the results presented here. Since the behaviour of the hybrid input-output algorithm depends on the particular image with which the algorithm is begun, it is essential to present a number of results, each obtained with a different starting image. This allows one to appreciate the algorithm's typical performance.

Figure 5.2 shows the error curves obtained, for four different starting images, when the hybrid input-output algorithm operates on the uncontaminated magnitude of the 128×128 pixel image shown in Figure 5.1. Five hundred iterations are performed in each case, with $\beta = 0.5$, and with both positivity and support constraints being enforced. Note the significant variation between the forms of the four error curves presented in Figure 5.2. The reconstructed images $\bar{f}_k[m,n]$ (as defined in §3.2.8.2) obtained after 100, 200, and 300 iterations, for the case in which the error curve is Figure 5.2a, are shown in Figure 5.3a,b,c respectively.

Upon examination of the reconstructed images shown in Figure 5.3, it is evident that the gross features of the image appear initially, being followed by the gradual emergence of fine detail. This suggests that the hybrid input-output algorithm recovers the phases of the low spatial frequency components first.

Note the plateaux in Figure 5.2b,c,d at which E_k^{I} remains for a significant number of iterations. These are due to the onset of stripe stagnation, as described in

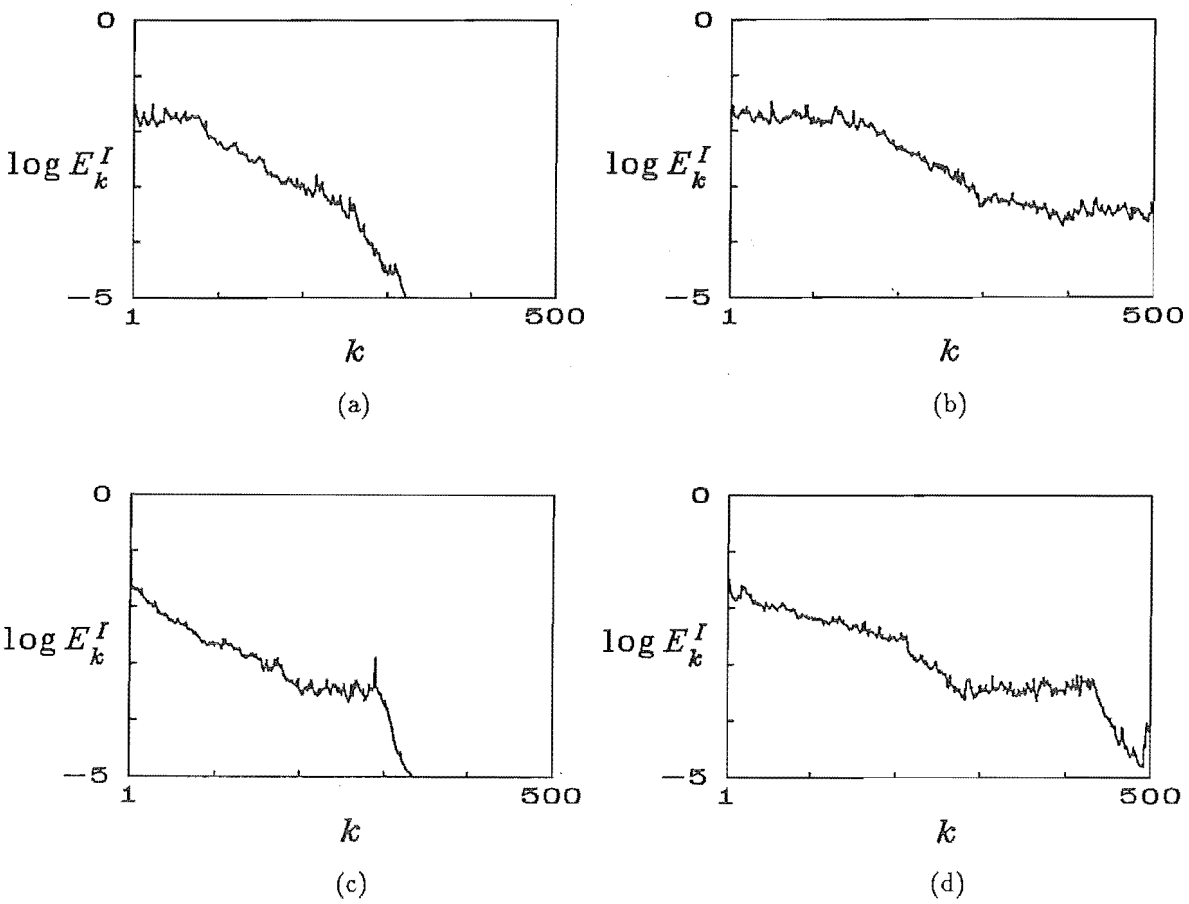


Figure 5.2 Error curves obtained, for four different starting images, when the hybrid input-output algorithm operates on the magnitude of the Fourier transform of the positive original image shown in Figure 5.1.

§3.2.8.3. Figure 5.3d shows the reconstructed image obtained after 400 iterations, for the case which resulted in the error curve of Figure 5.2d. Comparison with Figure 5.1 reveals that Figure 5.3d contains significant stripes. Although stripe stagnation is often overcome by performing further iterations (as in Figure 5.2c,d), it can sometimes be more computationally efficient to restart the algorithm with a different starting image. If two differently striped images are obtained, the techniques of Fienup and Wackerman (1986) may be employed to generate an improved image from the two striped images.

It is also necessary to demonstrate the performance of the hybrid input-output algorithm when it operates on the magnitude of an intricate image whose pixels can assume complex values. Figure 5.4 shows the error curves obtained, for four different starting images, when the hybrid input-output algorithm (with $\beta = 0.5$) operates on the uncontaminated Fourier magnitude of a 128×128 pixel complex image, the magnitude of which is that shown in Figure 5.1, and the phase of which is pseudo-randomly distributed in the range $[-\pi, \pi]$. Comparison of Figure 5.4 with Figure 5.2 confirms previous reports (Lane, 1987; Fienup, 1987) that the number of iterations required to recover a complex image is significantly greater than that required to

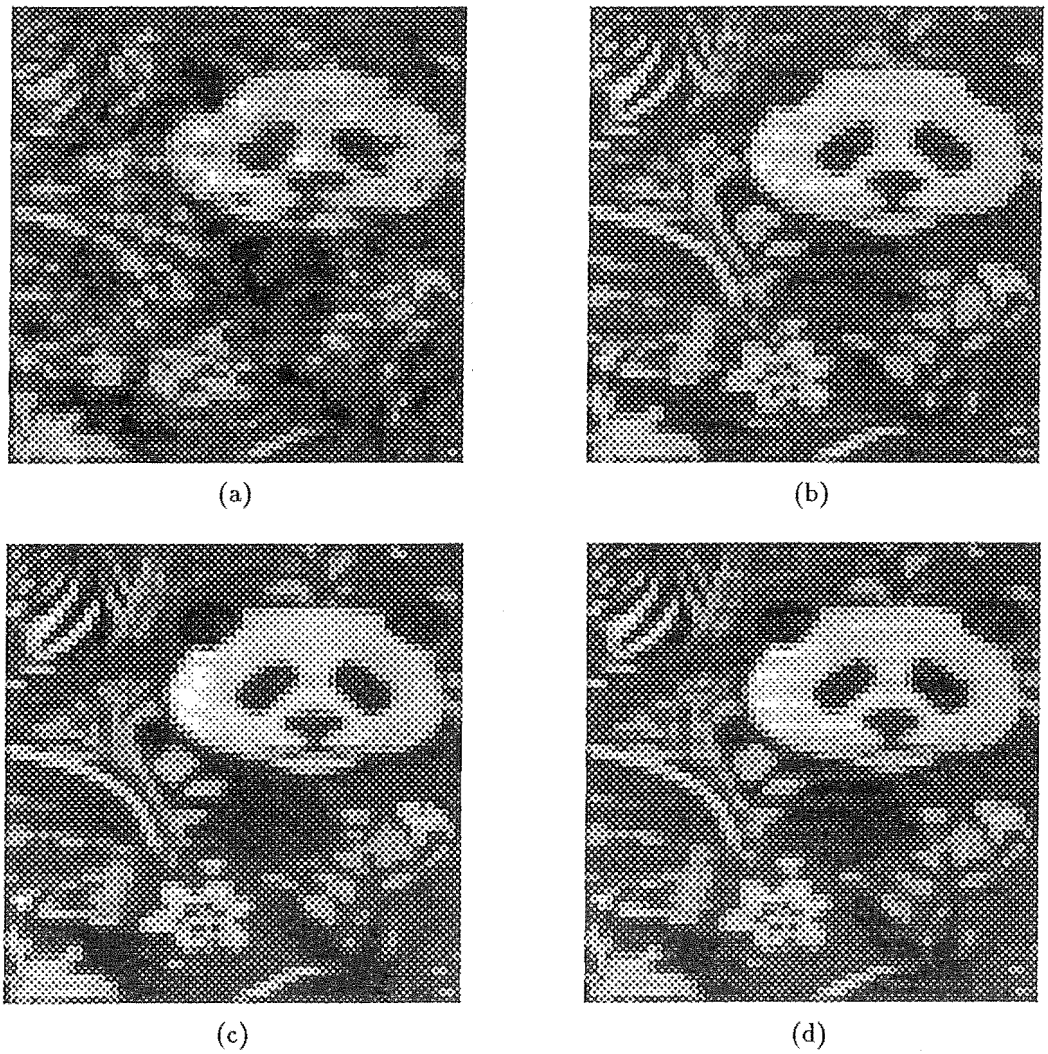


Figure 5.3 Reconstructed images corresponding to (a) $k = 100$, (b) $k = 200$, and (c) $k = 300$ on the error curve shown in Figure 5.2a, and (d) $k = 400$ on the error curve shown in Figure 5.2d.

recover a real image of the same size.

5.2.2 The Accelerated Hybrid Input-Output Algorithm

It was mentioned, in §5.2.1, that the hybrid input-output algorithm appears to first reconstruct the phases of low spatial frequency components. The inclusion of high spatial frequency components in the initial iterations is thus computationally wasteful. The basis of the algorithm introduced in this section, here termed the *accelerated hybrid input-output algorithm*, is to extract a fraction of the total number of given magnitude samples, and attempt to retrieve the phase corresponding to each of these extracted magnitude samples. A yet larger fraction of the given magnitude samples are then extracted, and the corresponding phases estimated, making use of the previously calculated phases. This procedure is repeated until all of the given magnitude samples have been employed. It is hoped that the amount of computation required to recover an image by this approach should be significantly less than that required

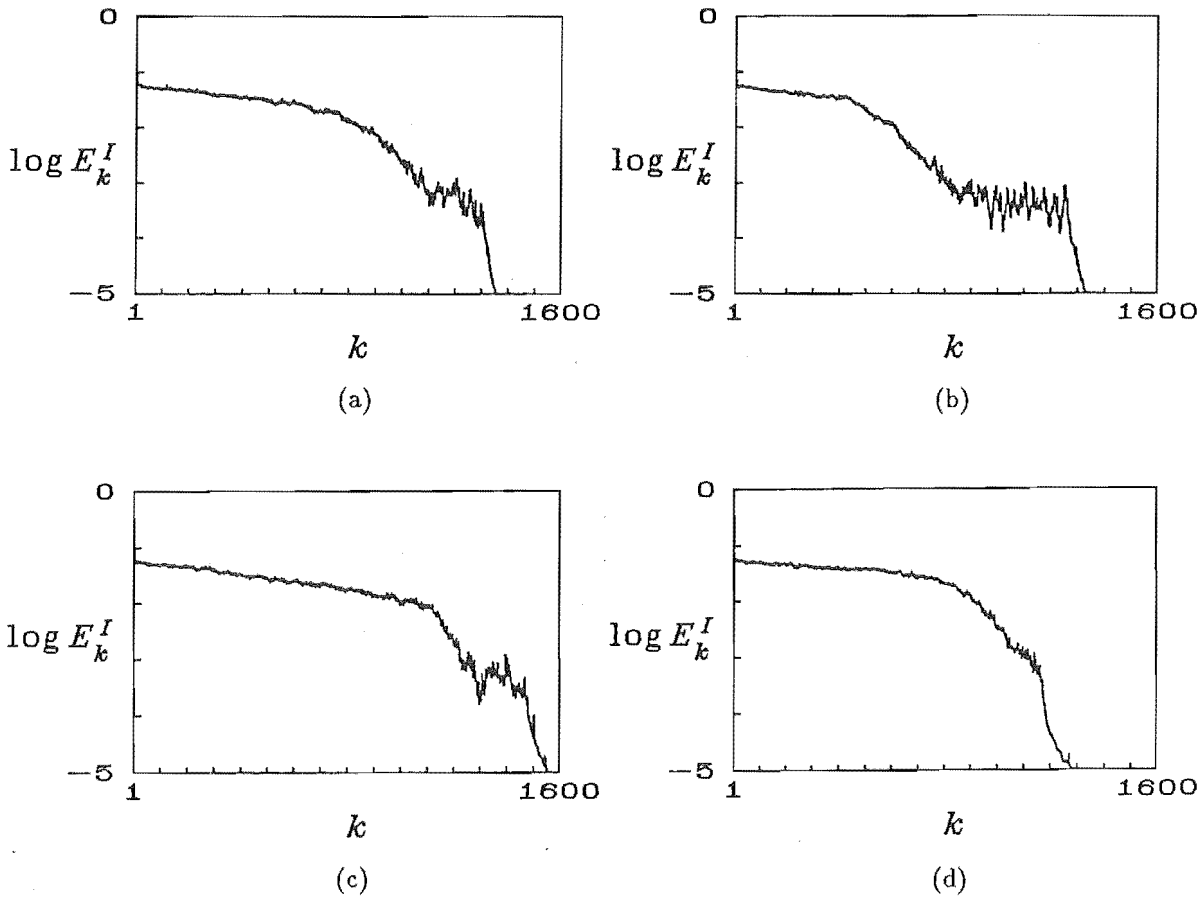


Figure 5.4 Error curves obtained, for four different starting images, when the hybrid input-output algorithm operates on the magnitude of the Fourier transform of the complex original image.

to recover the same image by the standard hybrid input-output algorithm.

Consider the discrete image $f[m,n]$ with spectrum $F[p,q]$, the (possibly contaminated) magnitude of which is $M[p,q]$. In order to reduce the number of samples in $M[p,q]$, it is necessary to multiply $M[p,q]$ by a window $W[p,q]$ which is zero outside some specified region. Only those samples of $M[p,q]$ which are inside the aforesaid region then need to be included in any computation.

The choice of $W[p,q]$ must be carefully considered. Evidently, $W[p,q]$ must be real and positive, to ensure that $W[p,q]M[p,q]$ is indeed the magnitude of some quantity. It is also desirable for $W[p,q]$ to be close to unity at as many samples as possible, so that the majority of the extracted magnitude samples are little distorted. However, the effect that $W[p,q]$ has in image space must also be considered. Since $W[p,q]$ is positive, it follows that

$$W[p,q]\,M[p,q] = \Big|W[p,q]\,F[p,q]\Big| \tag{5.1}$$

It is convenient to define the *equivalent windowed image* $f_w[m,n]$, as

$$f_w[m,n] = f[m,n] \odot w[m,n] \tag{5.2}$$

where $w[m,n] \longleftrightarrow W[p,q]$. Note from (5.1) that $f_w[m,n]$ has Fourier magnitude $W[p,q]M[p,q]$. Thus when attempting to reconstruct an image from the magnitude

$M[p,q]$ windowed by $W[p,q]$, the best that one can hope to do is to recover an image-form of $f_w[m,n]$.

Recall from §2.4.2 that a phase problem may have multiple solutions if the original image happens to be a convolution. It is worth considering if multiple solutions exist when attempting to recover an image from the windowed magnitude $W[p,q]M[p,q]$, since $f_w[m,n]$ is indeed a convolution. Since both $f[m,n] \odot w[m,n]$ and $f[m,n] \odot w^*[-m,-n]$ possess Fourier magnitude $W[p,q]M[p,q]$, they both constitute potential solutions. However, since $W[p,q]$ is real, it follows that $w^*[-m,-n] = w[m,n]$. There is thus only one image-form that is compatible with the magnitude $W[p,q]M[p,q]$.

The manner in which the effective windowed image $f_w[m,n]$ is determined by $W[p,q]$ is clarified by (5.2). For the hybrid input-output algorithm to be successful in recovering $f_w[m,n]$ from $W[p,q]M[p,q]$, the effective support of $f_w[m,n]$ must be as small as possible. From (5.2), it then follows that $w[m,n]$ ought to possess a main lobe that is as narrow as possible, and sidelobes that are as small as possible. However, since the image and Fourier space requirements on the window $W[p,q]$ are conflicting, a compromise must be sought.

The two-dimensional window $W[p,q]$ is constructed from a one-dimensional window $W_{1d}[p]$ by

$$W[p,q] = W_{1d}[p]\, W_{1d}[q] \tag{5.3}$$

An alternative approach to (5.3) is to form a circularly symmetric $W[p,q]$ from the one-dimensional window $W_{1d}[p]$. A number of one-dimensional windows are described and compared by Harris (1978). The Hamming window (Harris, 1978, §V.D) is here chosen for $W_{1d}[p]$, due to its appropriate compromise between effective width in one domain, and main lobe width and side lobe level in the other domain.

It is now assumed that the available Fourier magnitude data comprises $N \times N$ samples. The accelerated algorithm operates on quantities comprising $N_r \times N_r$ samples, where N_r is the *reduced image size*, satisfying $N_r \leq N$. Since most routines for computing two-dimensional FFTs operate on data with dimensions which are a power of two, it is sensible to choose N_r to be a power of two. It is now convenient to introduce a notation to explicitly indicate the dimensions (in pixels) of an image. The notation $a^{b \times b}[m,n]$ is invoked to indicate that the discrete image $a[m,n]$ comprises $b \times b$ pixels. The available Fourier magnitude data are thus denoted by $M^{N \times N}[p,q]$, and the corresponding original image by $f^{N \times N}[m,n]$.

The steps of the accelerated hybrid input-output algorithm are now listed:

Algorithm 5.1

Step 1: Set $N_r = N_r^0$.

Step 2: Form $M^{N_r \times N_r}[p,q]$ by extracting central $N_r \times N_r$ pixel region from $M^{N \times N}[p,q]$.

Step 3: If $N_r = N_r^0$, then generate pseudo-random phase $\Psi^{N_r \times N_r}[p,q]$,
otherwise set $\Psi^{N_r \times N_r}[p,q]$ to $\Phi^{\frac{N_r}{2} \times \frac{N_r}{2}}[p,q]$ at central $\frac{N_r}{2} \times \frac{N_r}{2}$ pixels of $\Psi^{N_r \times N_r}[p,q]$ and set $\Psi^{N_r \times N_r}[p,q]$ pseudo-randomly elsewhere.

Step 4: Form window $W^{N_r \times N_r}[p,q]$. If $N_r = N$, then set all pixels of $W^{N_r \times N_r}[p,q]$ to unity.

Step 5: Form $\hat{f}_0^{N_r \times N_r}[m,n] = \mathrm{FT}^{-1}\{W^{N_r \times N_r}[p,q]\, M^{N_r \times N_r}[p,q]\, \exp(i\Psi^{N_r \times N_r}[p,q])\}$.

Step 6: Shift $\hat{f}_0^{N_r \times N_r}[m,n]$ to maximize its energy within the square image mask of extent L^{N_r} pixels.

Step 7: Perform K^{N_r} hybrid input-output iterations, with square image mask of extent L^{N_r} pixels, on magnitude $W^{N_r \times N_r}[p,q]\, M^{N_r \times N_r}[p,q]$, with starting image $\hat{f}_0^{N_r \times N_r}[m,n]$, thereby producing $\hat{f}^{N_r \times N_r}[m,n]$.

Step 8: Form $\Phi^{N_r \times N_r}[p,q] = \mathrm{Ph}\{\mathrm{FT}\{\hat{f}^{N_r \times N_r}[m,n]\}\}$.

Step 9: If $N_r = N$ then stop,
otherwise double N_r and return to step 2.

It is necessary to mention some further aspects of Algorithm 5.1. The initial reduced image size N_r^0 should be chosen such that the effective support of the reconstructed image $\hat{f}^{N_r^0 \times N_r^0}[m,n]$ is usefully smaller (so that the support constraint of the hybrid input-output algorithm has some effect) than $N_r^0 \times N_r^0$ pixels. Computational experience suggests that N_r^0 ought to be no less than 16.

The choice of K^{N_r} (the number of hybrid input-output iterations performed when operating on $N_r \times N_r$ samples of magnitude data) is also worth mentioning. It is computationally feasible for K^{N_r}, when $N_r < N$, to be larger than K^N, since one iteration of the hybrid input-output algorithm in which the full $N \times N$ pixels of magnitude data are employed is equally computationally demanding as approximately $[N \log_2 N/(N_r \log_2 N_r)]^2$ iterations on $N_r \times N_r$ samples of magnitude data. Table 5.1 summarizes the computational savings which result from employing various values of N_r.

N_r	N_{eq}
16	1960
32	314
64	5
128	1

Table 5.1 Equivalent number of hybrid input-output iterations N_{eq} which, for various values of reduced image size N_r, require the same computational expenditure as one iteration with $N = 128$.

If the original image is real, it should be ensured, when performing Step 3 of Algorithm 5.1, that $\Psi^{N_r \times N_r}[p,q]$ possesses odd symmetry. Step 6 ensures that $\hat{f}_0^{N_r \times N_r}[m,n]$ is appropriately located within its image mask. This step is necessary since the relative size of the image mask (i.e. L^{N_r}/N_r) should decrease each time that N_r is doubled.

Computational examples of the accelerated hybrid input-output algorithm are now presented.

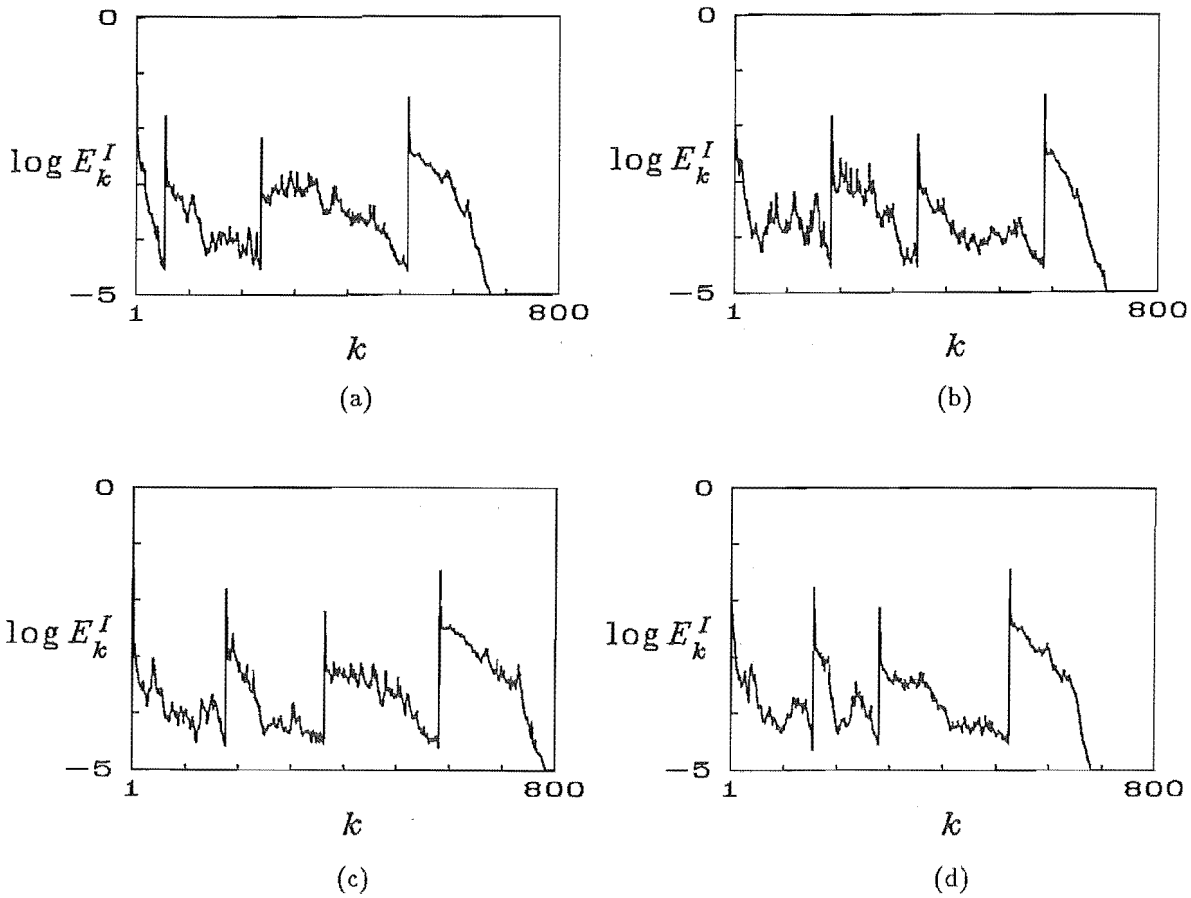


Figure 5.5 Error curves obtained, for four different starting images, when the accelerated hybrid input-output algorithm operates on the magnitude of the Fourier transform of the positive original image.

5.2.3 Computational Examples

The performance of the accelerated hybrid input-output algorithm, as described in §5.2.2, is now demonstrated. For the first example, the algorithm is applied to the 128×128 sample magnitude of the Fourier transform of the positive image shown in Figure 5.1. In step 7 of Algorithm 5.1, a positivity constraint is applied in conjunction with the support constraint. The values of L^{N_r}, the extent of the image mask when operating on $N_r \times N_r$ samples of magnitude data, are chosen to be $L^{16} = 11$, $L^{32} = 19$, $L^{64} = 35$, and $L^{128} = 64$. Each value of K^{N_r}, the number of iterations performed when operating on $N_r \times N_r$ samples of magnitude data, is simply that which is required before E_k^I falls below a preset level. This level is set to 3×10^{-5} for $N_r = 16$, 32, and 64, and to 10^{-5} for $N_r = 128$.

The error curves obtained when the accelerated hybrid input-output algorithm operates as described above, for four different starting phases, are shown in Figure 5.5. The values of k at which E_k^I exhibits a sudden pronounced increase, in each of the error curves in Figure 5.5, are those values at which N_r is doubled. This increase in E_k^I is due to the incorporation of high spatial frequency components with incorrect phases,

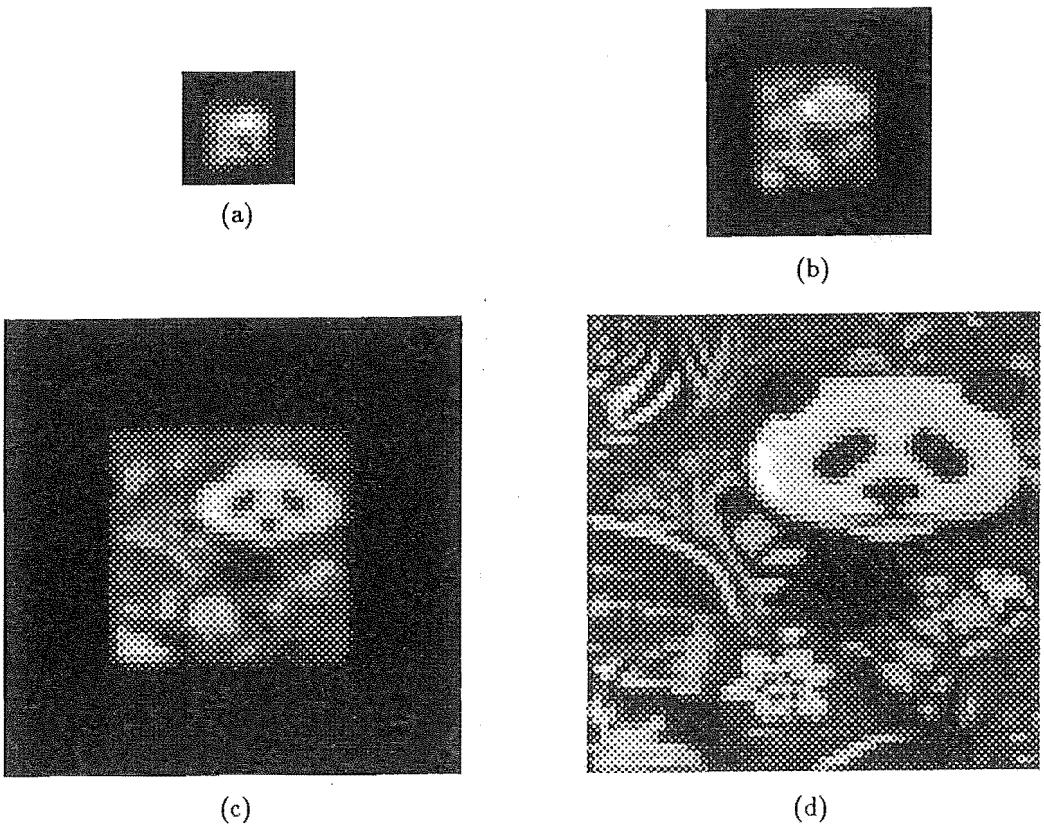


Figure 5.6 Reconstructed images corresponding to the error curve shown in Figure 5.5a. (a) $\hat{f}^{16\times16}[m,n]$, (b) $\hat{f}^{32\times32}[m,n]$, (c) $\hat{f}^{64\times64}[m,n]$, and (d) central 64×64 pixel region of $\hat{f}^{128\times128}[m,n]$.

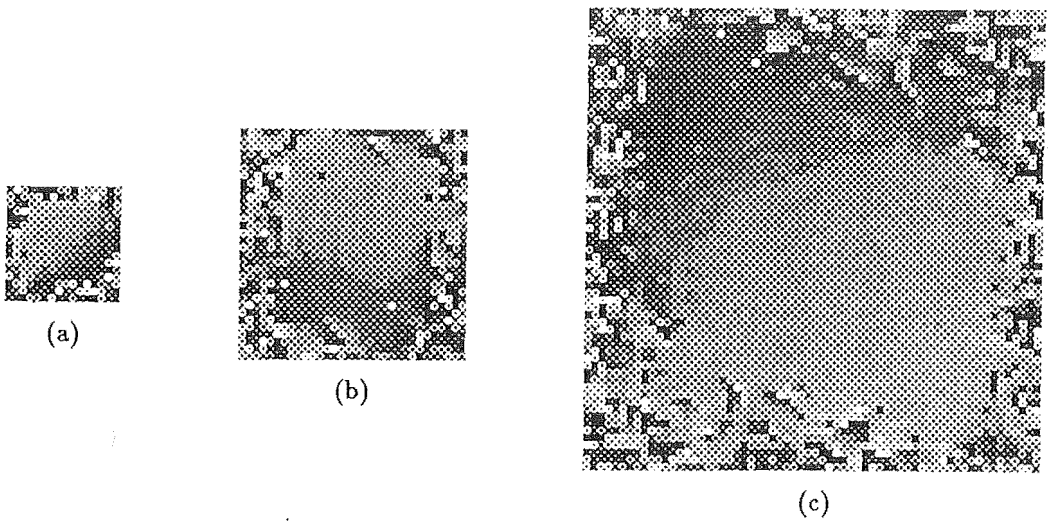


Figure 5.7 Phase differences, computed modulo 2π and displayed such that $-\pi$ and π correspond to black and white respectively, between spectra of original image and (a) $\hat{f}^{16\times16}[m,n]$, (b) $\hat{f}^{32\times32}[m,n]$, and (c) $\hat{f}^{64\times64}[m,n]$ as shown in Figure 5.6.

Number of iterations of the standard hybrid input-output algorithm	Effective number of iterations of the accelerated hybrid input-output algorithm
323	214
> 500	169
335	250
> 500	207

Table 5.2 Summary of the computational requirements for the examples presented in Figure 5.2 and Figure 5.5.

which result in significant energy appearing outside the image mask. Figure 5.6 shows $\hat{f}^{16\times16}[m,n]$, $\hat{f}^{32\times32}[m,n]$, $\hat{f}^{64\times64}[m,n]$, and $\hat{f}^{128\times128}[m,n]$ for the case in which the error curve is Figure 5.5a. Note how the image detail improves as N_r is increased. Figure 5.7 shows the differences (modulo 2π) between the phase of the original image and $\Phi^{16\times16}[p,q]$, $\Phi^{32\times32}[p,q]$, and $\Phi^{64\times64}[p,q]$ respectively. This demonstrates that each $\Phi^{N_r\times N_r}[p,q]$ is recovered correctly (apart from a linear phase shift, due to a translation of the image) throughout a region of Fourier space which extends almost to the edge of the window $W^{N_r\times N_r}[m,n]$.

The performance of the accelerated hybrid input-output algorithm can be compared with that of the standard hybrid input-output algorithm by comparing the error curves in Figure 5.5 and Figure 5.2, since the same original image is employed in both cases. However, direct comparison of the computational effort expended by the two algorithms is difficult, since the computational effort per iteration of the accelerated hybrid input-output algorithm increases with N_r, as explained in §5.2.2. To aid comparison, Table 5.2 lists the number of iterations of the standard hybrid input-output algorithm (for the four cases presented in Figure 5.2), and the effective number of 128×128 iterations performed by the accelerated hybrid input-output algorithm (for the four cases presented in Figure 5.5), that are required to ensure that $E_k^{\mathrm{I}} < 10^{-5}$. The computational savings effected by the accelerated hybrid input-output algorithm are clearly significant.

The second example of the operation of the accelerated hybrid input-output algorithm concerns the reconstruction of a complex image. The algorithm is applied to the 128×128 sample magnitude of the Fourier transform of the same complex image as that employed in §5.2.1. The extents of the image mask employed in Step 7 of Algorithm 5.1 are chosen to be $L^{16} = 10$, $L^{32} = 18$, $L^{64} = 34$, and $L^{128} = 64$. The values of K^{N_r} employed in Step 7 are $K^{16} = 2000$, $K^{32} = 2000$, $K^{64} = 1000$, with K^{128} being that required to ensure that $E_k^{\mathrm{I}} < 10^{-5}$. The error curves thus obtained, for four different starting phases, are shown in Figure 5.8. The computational requirements of these examples are summarized in Table 5.3

5.2.4 Discussion

The first example presented in §5.2.3 demonstrates that the accelerated hybrid input-output algorithm affords a significant computational saving over the standard hybrid input-output algorithm when reconstructing positive images. The second example,

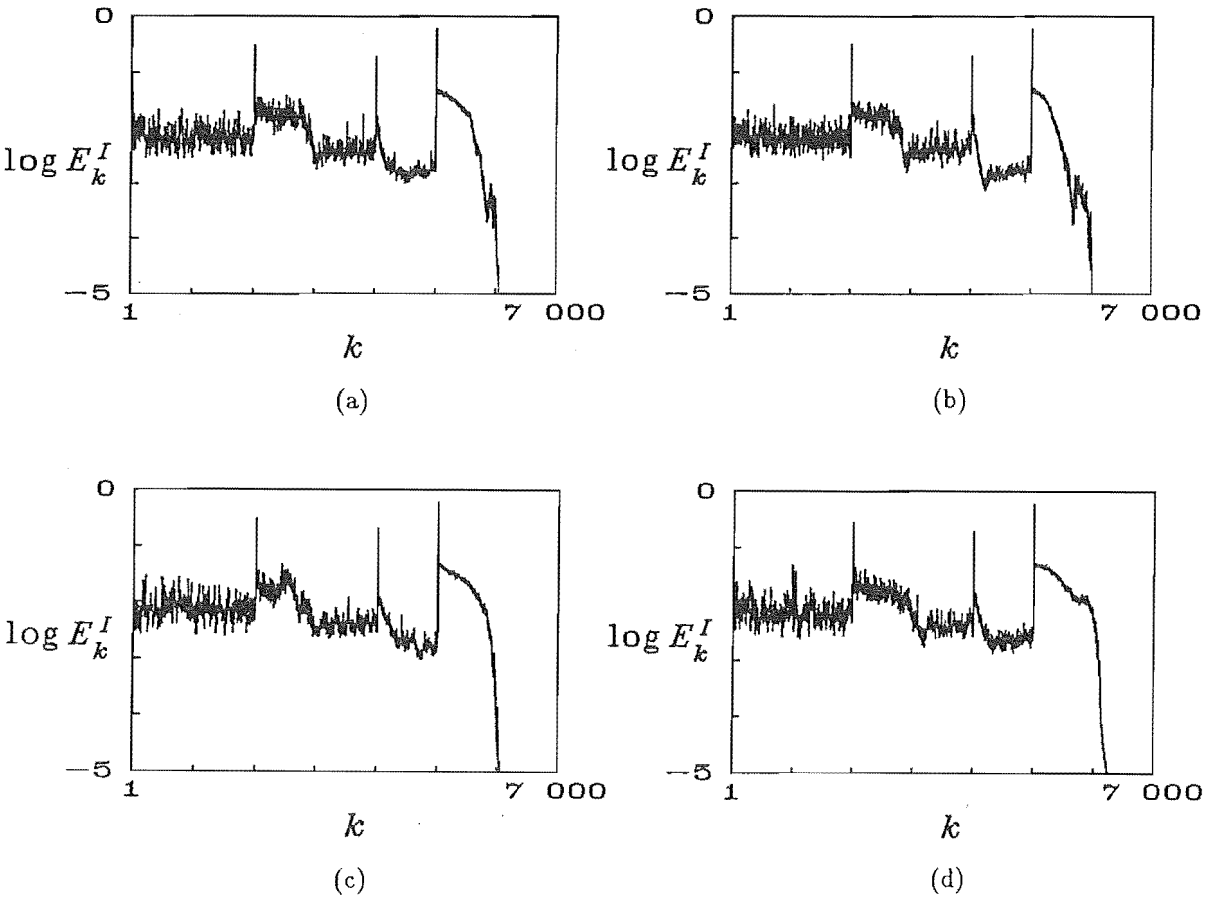


Figure 5.8 Error curves obtained, for four different starting images, when the accelerated hybrid input-output algorithm operates on the magnitude of the Fourier transform of the complex original image.

Number of iterations of the standard hybrid input-output algorithm	Effective number of iterations of the accelerated hybrid input-output algorithm
1357	1331
1332	1285
1555	1315
1284	1496

Table 5.3 Summary of the computational requirements for the examples presented in Figure 5.4 and Figure 5.8.

however, suggests that the accelerated algorithm has little advantage over the standard algorithm when recovering a complex image. It appears that this difference in effectiveness is due to the different natures of the Fourier magnitudes used for these two examples. For the first example, in which the original image is positive, the majority of the energy of the spectrum is concentrated near to the origin in Fourier space. Consequently, the starting image $\hat{f}_0^{128\times128}[m,n]$ (as computed in Step 4 of Algorithm 5.1) is "close" to the original image $f^{128\times128}[m,n]$ (provided that $\Phi^{64\times64}[p,q]$

is a faithful estimate of the true phase), and so the convergence from $\hat{f}_0^{128\times128}[m,n]$ to the true solution ought to be rapid. For the second example, in which the original image is complex, the energy of the spectrum is spread roughly uniformly throughout the 128×128 sample Fourier magnitude. So even if $\Phi^{64\times64}[p,q]$ is a faithful estimate of the true phase, $\hat{f}_0^{128\times128}[m,n]$ tends to be significantly different from $f^{128\times128}[m,n]$, which results in slow convergence from $\hat{f}_0^{128\times128}[m,n]$ to the true solution.

The value of K^{N_r} (as defined in §5.2.2) employed in Step 7 of Algorithm 5.1, should be chosen with care. It is desirable to perform sufficient iterations to drive E_k^I as low as possible, thereby ensuring that $\Phi^{N_r\times N_r}[p,q]$ is as faithful as possible. However, there is always some value, determined by both the contamination level and the form of $w^{N_r\times N_r}[m,n]$, below which E_k^I is incapable of falling.

The choice of L^{N_r} (as defined in §5.2.2 employed in Step 7 of Algorithm 5.1 is also an important consideration. Sensible values for L^{N_r} can be derived from an estimate of the support of the original image, together with knowledge of the form of $w^{N_r\times N_r}[m,n]$. If L^{N_r} is too large, then slow convergence of the hybrid input-output iterations results (especially when reconstructing a complex image). Conversely, if L^{N_r} is too small, the hybrid input-output iterations are prevented from converging to a reasonable solution, and thus results in a poor $\Phi^{N_r\times N_r}[p,q]$.

It should be pointed out that it is possible for stagnation to occur during the hybrid input-output iterations in Step 7 of Algorithm 5.1. In such a situation, it may well be computationally more efficient to return to Step 3, rather than to perform an excessive number of iterations. Such stagnation may well be avoided by gradually increasing (as opposed to doubling, as is done in Algorithm 5.1) the size of the window $W[p,q]$. By doing so, only a small number of new phases must be estimated following each increase in the window size, and so stagnation would presumably be less likely to occur. However, the required computational effort may well increase.

Windows other than the Hamming window could, of course, be employed for $W^{N_r\times N_r}[p,q]$ in Step 4 of Algorithm 5.1. A window $W^{N_r\times N_r}[p,q]$ which has low side lobe levels in image space results in a reconstructed image $\hat{f}^{N_r\times N_r}[m,n]$ with little energy outside the image mask (provided that L^{N_r} is chosen appropriately). However, if the Fourier magnitude data are appreciably contaminated, then $\hat{f}^{N_r\times N_r}$ must necessarily possess appreciable energy outside the image mask. In such a situation, there is little point in employing a window which has extremely low sidelobe levels in image space. It may well be advantageous to adjust $W^{N_r\times N_r}[p,q]$ in accordance with the contamination level. A Dolph-Chebyshev window (Harris, 1978, §V.H) would then be convenient, since its sidelobe level can be easily adjusted.

5.3 An Alternative Input-Output Algorithm

The rationale underlying the class of iterative phase retrieval algorithms known as input-output algorithms is reviewed in §3.2.8.2. Furthermore, two members of this class, namely the basic input-output algorithm and the hybrid input-output algorithm, are described in detail in §3.2.8.2. The hybrid input-output algorithm is the most commonly employed phase retrieval algorithm at present (e.g. Dainty and Fienup, 1987; Bates and McDonnell, 1986, §23). Indeed, it is invoked in Chapter 4

and §5.2 of this thesis. Despite its popularity, the hybrid input-output algorithm is not always the most effective phase retrieval technique, as is demonstrated in this section.

A new alternative input-output algorithm is described in §5.3.1, which is in fact derived by appropriately combining the basic input-output algorithm with the hybrid input-output algorithm. The performance of this new algorithm is illustrated in §5.3.2. The algorithm is discussed further in §5.3.3.

5.3.1 The Modified Input-Output Algorithm

Recall, from §3.2.8.2, the rationale behind the input-output algorithms. The first three steps of the error-reduction algorithm (3.49a-c) are viewed as a nonlinear system, with input $\tilde{f}_k[m,n]$ and output $\bar{f}_k[m,n]$ at the k^{th} iteration. In each of the various input-output algorithms, $\tilde{f}_k[m,n]$ is adjusted in an attempt to produce a $\bar{f}_k[m,n]$ which complies with the appropriate image space constraints (e.g. positivity and/or support). The hybrid input-output algorithm is a combination of the error-reduction and the basic input-output algorithms, and often proves extremely effective because it cannot stagnate in the same manner as the error-reduction algorithm can (as described in detail in §3.2.8.2). However, the reasoning on which the hybrid input-output algorithm was originally based appears to be somewhat *ad hoc* (Fienup, 1982). It is thus worthwhile to seek alternative input-output algorithms.

For applications in which positivity is the sole constraint which can be enforced in image space, such as crystallography (§2.5.1) and some areas of microscopy (e.g. Bates and Tan, 1986), it appears that the hybrid input-output algorithm is not ideal, for the following reason. Consider a particular pixel in the output $\bar{f}_k[m,n]$ located at $[m',n']$, such that $\bar{f}_k[m',n']$ happens to be negative. As described in §3.2.8.2, the value of the input pixel $\tilde{f}_{k+1}[m',n']$ is obtained by increasing $\tilde{f}_k[m',n']$ by an amount $-\beta\bar{f}_k[m',n']$. In a similar manner, $\tilde{f}[m',n']$ is increased at each iteration until eventually (say at the j^{th} iteration) $\bar{f}_j[m',n']$ becomes positive. Once this occurs, the new input $\tilde{f}_{j+1}[m',n']$ is promptly set to $\bar{f}_j[m',n']$, irrespective of the value of $\tilde{f}_j[m',n']$. The basic input-output algorithm, on the other hand, sets $\tilde{f}_{j+1}[m',n']$ to $\tilde{f}_j[m',n']$ once $\bar{f}_j[m',n']$ becomes positive, which seems to be a more sensible course of action. However, the basic input-output algorithm unfortunately does not converge, for the reason stated in §3.2.8.2.

The above paragraph suggests that an appropriate combination of the basic and hybrid input-output algorithms may exhibit the desirable characteristics of each algorithm. Specifically, consider the algorithm specified by (3.49) with (3.49d) replaced by

$$\tilde{f}_{k+1}[m,n] = \gamma\,\tilde{f}^{\text{bio}}_{k+1}[m,n] + (1-\gamma)\,\tilde{f}^{\text{hio}}_{k+1}[m,n] \tag{5.4}$$

where γ is a constant satisfying $0 \leq \gamma \leq 1$, and $\tilde{f}^{\text{bio}}_{k+1}[m,n]$ and $\tilde{f}^{\text{hio}}_{k+1}[m,n]$ are the inputs $\tilde{f}_{k+1}[m,n]$ as defined by (3.58) and (3.59) for the basic and hybrid input-output algorithms respectively. This new algorithm is here termed the *modified input-output algorithm*. Note that the modified input-output algorithm with $\gamma = 0$ is equivalent to the hybrid input-output algorithm, while with $\gamma = 1$ it is equivalent to the basic

input-output algorithm. Substituting (3.58) and (3.59) into (5.4) yields

$$\tilde{f}_{k+1}[m,n] = \begin{cases} \gamma\, \tilde{f}_k[m,n] + (1-\gamma)\, \bar{f}_k[m,n] & \text{if } [m,n] \notin V_k \\ \tilde{f}_k[m,n] - \beta\, \bar{f}_k[m,n] & \text{if } [m,n] \in V_k \end{cases} \tag{5.5}$$

where β is the hybrid input-output gain parameter, and V_k is the set of pixels at which $\bar{f}_k[m,n]$ violates the image space constraints. One iteration of the modified input-output algorithm is then implemented by performing (3.49a-c) followed by (5.5).

The performance of the modified input-output algorithm is now demonstrated.

5.3.2 Computational Examples

As is evident from (5.5), the modified input-output algorithm has two parameters which may be varied, namely β and γ. It is thus necessary to provide results of the operation of the algorithm for a range of values of β and γ. Furthermore, to gain an appreciation of the typical performance of the modified input-output algorithm, it is essential to repeat each trial with a number of different starting images.

As mentioned in §5.4.1, the modified input-output algorithm is expected to be particularly superior to the hybrid input-output algorithm in situations in which only a positivity constraint is employed. Consequently, the examples presented here are those obtained when the modified input-output algorithm operates upon the magnitude of a non-oversampled Fourier transform, specifically that of the positive 32×32 pixel image shown in Figure 5.9. Note, in particular, that the Fourier magnitude data $M[p,q]$ only comprise 32×32 pixels, since the original image has not been "packed with zeros".

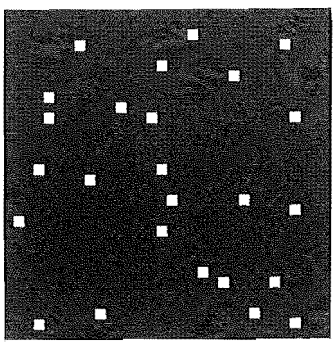


Figure 5.9 Original 32×32 pixel positive image.

The modified input-output algorithm is run ten times, with a different pseudo-random starting image employed each time, for each combination of β and γ. Twenty different combinations of β and γ are tested, namely $\beta = 0.25, 0.5, 0.75, 1.0$ and $\gamma = 0, 0.25, 0.5, 0.75, 1.0$. None of the trials for which $\gamma = 1.0$ converged. This is not surprising, since the modified input-output algorithm with $\gamma = 1.0$ is equivalent to the basic input-output algorithm, which cannot be expected to converge for the reasons stated in §3.2.8.2. Furthermore, none of the trials for which $\beta = 1.0$ converged.

For the remaining trials, it is inconvenient to display all 120 error curves. Instead, the quantity NC_k, which is the number of trials (out of a possible 10) which have converged after k iterations, is plotted in Figure 5.10 for each combination of β and γ. A trial is here deemed to have converged once $E_k^{\mathrm{I}} < 10^{-5}$. The results presented in Figure 5.10 are discussed further in §5.3.3.

5.3.3 Discussion

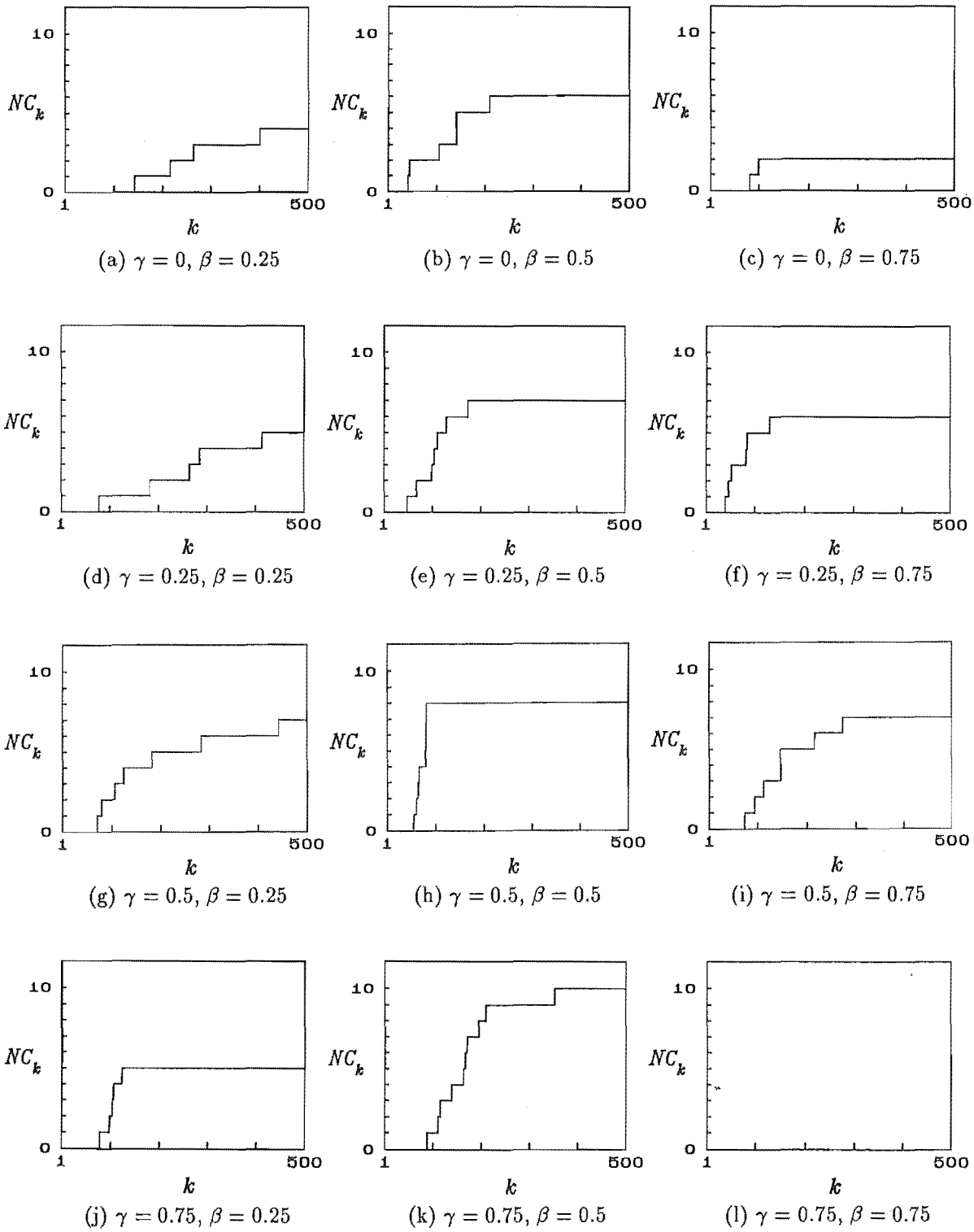
When comparing each of the results presented in Figure 5.10, the important characteristics to bear in mind are the number of trials which converge after a "small" number of iterations, as well as the total number of trials which converge within 500 iterations. Although some of the trials do not converge within 500 iterations, in all but one of the results presented in Figure 5.10, it is computationally wasteful to continue iterating until convergence is reached. Instead, it would be more computationally efficient to restart the algorithm with a different starting image.

It is worthwhile comparing the performance of the modified input-output algorithm with that of the hybrid input-output algorithm. Since the modified input-output algorithm with $\gamma = 0$ is equivalent to the hybrid input-output algorithm, the results presented in Figure 5.10a,b,c are simply those which would be obtained by invoking the hybrid input-output algorithm with $\beta = 0.25, 0.5$ and 0.75 respectively. Comparison of Figure 5.10a–c with Figure 5.10d–l reveals that for each of $\beta = 0.25, 0.5, 0.75$ there is some non-zero γ for which the modified input-output algorithm outperforms the hybrid input-output algorithm. Furthermore, it is evident that the best performance (of the trials performed here) of the modified input-output algorithm is that for $\beta = 0.5$ and $\gamma = 0.5$, whilst the best performance of the hybrid input-output algorithm is that for $\beta = 0.5$. Comparison of Figure 5.10h with Figure 5.10b reveals that the best performance of the modified input-output algorithm is a significant improvement over that of the hybrid input-output algorithm.

It is evident from Figure 5.10 that, for $\beta = 0.25, 0.5$ and 0.75, the modified input-output algorithm performs well with $\gamma = 0.75, 0.5$ and 0.25 respectively. This suggests that, for a particular β, an appropriate choice of γ is $\gamma = 1 - \beta$. However, there is no obvious reason why this should be so.

The example presented in §5.3.2 clearly demonstrates the superiority of the modified input-output algorithm over the hybrid input-output algorithm for applications in which only a positivity constraint can be employed. Further trials have shown that the modified input-output algorithm has no advantage over the hybrid input-output algorithm when both support and positivity constraints are employed. This is not surprising, since any pixels of $\bar{f}_k[m, n]$ lying outside the image mask always violate the image space constraints. Thus the situation described in §5.3.1, in which an output pixel that violates the constraints at one iteration subsequently meets the constraints at the next iteration, cannot occur for those pixels lying outside the image mask. This situation may occur for pixels within the image mask, but the majority of such pixels typically assume positive values after a small number of iterations.

The example presented in §5.3.2 is for a rather simple image. However, iterative phase retrieval algorithms of the type invoked here appear to be incapable of reconstructing images of significantly greater intricacy than that of Figure 5.9, unless



(a) $\gamma = 0, \beta = 0.25$ (b) $\gamma = 0, \beta = 0.5$ (c) $\gamma = 0, \beta = 0.75$

(d) $\gamma = 0.25, \beta = 0.25$ (e) $\gamma = 0.25, \beta = 0.5$ (f) $\gamma = 0.25, \beta = 0.75$

(g) $\gamma = 0.5, \beta = 0.25$ (h) $\gamma = 0.5, \beta = 0.5$ (i) $\gamma = 0.5, \beta = 0.75$

(j) $\gamma = 0.75, \beta = 0.25$ (k) $\gamma = 0.75, \beta = 0.5$ (l) $\gamma = 0.75, \beta = 0.75$

Figure 5.10 Results obtained when the modified input-output algorithm operates on the magnitude of the Fourier transform of the image shown in Figure 5.9, with the indicated values of γ and β.

further *a priori* information is available.

It is worth noting that the error-reduction algorithm, along with the basic, hybrid, and modified input-output algorithms, can be described by (3.49) with (3.49d) replaced by

$$\tilde{f}_{k+1}[m,n] = \begin{cases} c_1\,\tilde{f}_k[m,n] + c_2\,\bar{f}_k[m,n] & \text{if } [m,n] \notin V_k \\ c_3\,\tilde{f}_k[m,n] + c_4\,\bar{f}_k[m,n] & \text{if } [m,n] \in V_k \end{cases} \tag{5.6}$$

where the constants c_1, c_2, c_3, c_4 are listed in Table 5.4.

Algorithm Name	c_1	c_2	c_3	c_4
error-reduction algorithm	0	1	0	0
basic input-output algorithm	1	0	1	$-\beta$
hybrid input-output algorithm	0	1	1	$-\beta$
modified input-output algorithm	γ	$1-\gamma$	1	$-\beta$

Table 5.4 The parameters of (5.6) which describe various iterative phase retrieval algorithms.

Both the hybrid and the modified input-output algorithms have been devised by somewhat *ad hoc* means. It is thus possible that new and improved input-output algorithms could well be derived, which follow the general form given in the preceding paragraph, but which assume different values of c_1, c_2, c_3 and c_4 than those listed in Table 5.4.

5.4 An Alternative Implementation of Simulated Annealing for Phase Retrieval

The general simulated annealing algorithm, for effecting the minimization of a multimodal function, is described in §3.1.2.2. The application of the simulated annealing method to the Fourier phase problem by Nieto-Vesperinas *et al.* (1988) is described in §3.2.9.2, and mention is made, in §5.1, of the computational requirements of this algorithm. A new strategy for applying the simulated annealing method to the Fourier phase problem is introduced in this section. This implementation operates in Fourier space, in contrast to that of Nieto-Vesperinas *et al.* (1988), which operates in image space. It appears that this implementation may be more computationally efficient, when reconstructing complex images, than that of Nieto-Vesperinas *et al.* (1988).

The new simulated annealing algorithm for phase retrieval is detailed in §5.4.1, with computational examples being presented in §5.4.2. Further details of this algorithm are discussed in §5.4.3.

5.4.1 Algorithm Description

A number of phase retrieval methods have been described, in §3.2.5, which are based on the sampling theorem. So too is the phase retrieval algorithm about to be described, which is here called the *Fourier space simulated annealing algorithm.* Recall from §3.2.5.1 that, when attempting to solve a Fourier phase problem, significantly

more than half of the total number of available Fourier magnitude samples must be employed in order to ensure that a unique solution is obtained. To solve the Fourier phase problem by such methods, a set of nonlinear equations must be solved. Unfortunately, the cost function associated with this set of equations is usually multimodal if more than half of the total number of available Fourier magnitude samples are employed. This point has been explained in greater detail in §3.2.5. Since the simulated annealing method is known to be effective for minimizing multimodal functions, it seems that a unique solution to the Fourier phase problem might be obtained by minimizing the abovementioned cost function by the simulated annealing method. This idea is manifested in the algorithm which is now described.

Consider a discrete image $f[m,n]$, comprising $N \times N$ pixels, which is zero outside the central $\frac{N}{2} \times \frac{N}{2}$ pixel region. The DFT of $f[m,n]$, denoted by $F[p,q]$, thus also comprises $N \times N$ samples. On defining $M[p,q]$ and $\Phi[p,q]$ by $F[p,q] = M[p,q]\exp(i\,\Phi[p,q])$, it is seen that the data for the Fourier phase problem are the N^2 values of $M[p,q]$. Now since $f[m,n]$ is zero outside the central $\frac{N}{2}\times\frac{N}{2}$ pixel region, it follows from the sampling theorem (§1.5) that only every second sample, in the p and q directions, of $F[p,q]$ are required to adequately represent $F[p,q]$ (and hence $f[m,n]$). Such samples correspond to the Nyquist samples described in §1.5. It is now assumed, for convenience, that the Nyquist samples of $F[p,q]$ are those samples for which p and q are both even. It then follows that the in-row, in-column, and diagonal in-between sample points (as introduced in §3.2.5.1) are respectively those sample points for which p is odd and q is even, p is even and q is odd, and p and q are both odd.

If some estimate $\hat{\Phi}[p,q]$ of $\Phi[p,q]$ at the Nyquist sample points is available, then the corresponding estimate $\hat{F}[p,q]$ of $F[p,q]$, as provided by the sampling theorem (§1.5), is given by

$$\hat{F}[p,q] = \sum_{k=0}^{\frac{N}{2}-1} \sum_{l=0}^{\frac{N}{2}-1} M[2k,2l]\,\exp(i\,\hat{\Phi}[2k,2l])\,\mathrm{sinc}(p-2k)\,\mathrm{sinc}(q-2l) \tag{5.7}$$

Note that $|\hat{F}[p,q]|$ only differs from $M[p,q]$ at the in-between sampling points. As is the case for the other sampling theorem based methods mentioned in §3.2.5, one attempts to solve the Fourier phase problem by choosing a $\hat{\Phi}[p,q]$ such that $|\hat{F}[p,q]| = M[p,q]$ at the in-between sampling points.

It is necessary to define a cost function which serves as a measure of the misfit between $|\hat{F}[p,q]|$ and $M[p,q]$. But recall from §3.2.5.2 that it appears to be sufficient to ensure that $|\hat{F}[p,q]| = M[p,q]$ at only 3/4 of the total number of available magnitude samples. Since $|\hat{F}[p,q]| = M[p,q]$ at the Nyquist sample points, by virtue of (5.7), it is sufficient (and computationally advantageous, as explained in the following paragraph) for the cost function to reflect the misfit between $|\hat{F}[p,q]|$ and $M[p,q]$ at only the in-row and in-column in-between sample points. The cost function $Q\{\hat{\Phi}[p,q]\}$ is here defined as

$$Q\{\hat{\Phi}[p,q]\} = \sum_{\substack{p,q\,:\,p+q \text{ is odd} \\ 0\le p,q\le N-1}} \sum \left(|\hat{F}[p,q]| - M[p,q]\right)^2 \Big/ M[p,q]^2 \tag{5.8}$$

where $\hat{F}[p,q]$ is defined by (5.7). The Fourier phase problem is then solved by minimizing $Q\{\hat{\Phi}[p,q]\}$ with the aid of the simulated annealing method.

To explain the computational savings resulting from the omission of the diagonal in-between samples in the computation of $Q\{\hat{\Phi}[p,q]\}$, it is necessary to consider making a perturbation to $\hat{\Phi}[p,q]$ at a particular Nyquist sample point $[p',q']$, thereby producing $\Phi'[p,q]$. It is evident from (5.7) that such a perturbation affects the value of $\hat{F}[p,q]$ at every diagonal in-between sample point (of which there are $N^2/4$), at in-row in-between sample points for which $q = q'$ (of which there are $N/2$), and at in-column in-between sample points for which $p = p'$ (of which there are $N/2$). The computation of ΔQ, as defined by

$$\Delta Q = Q\{\Phi'[p,q]\} - Q\{\hat{\Phi}[p,q]\} \tag{5.9}$$

thus involves $o(N)$ operations. However, if the diagonal in-between samples were to be included in $Q\{\hat{\Phi}[p,q]\}$, then the computation of ΔQ would involve $o(N^2)$ operations.

The steps of the Fourier space simulated annealing algorithm are now listed:

Algorithm 5.2

Step 1: Repeat the following two steps, which together constitute one cycle, for $n_c = 1$ to N_c:

- Step 1.1: Compute temperature T and scale of perturbations α.
- Step 1.2: Repeat the following step, which constitutes one scan, for $n_s = 1$ to N_s:
  - Step 1.2.1: Repeat the following three steps for all even values of p' and q' between 0 and $N-2$:
    - Step 1.2.1.1: Perturb $\hat{\Phi}[p,q]$ at $[p',q']$, by an amount chosen pseudo-randomly in the range $[-\alpha,\alpha]$, thereby producing $\Phi'[p,q]$.
    - Step 1.2.1.2: Calculate ΔQ as defined by (5.9).
    - Step 1.2.1.3: If $\Delta Q \leq 0$, accept the perturbation.
    - Step 1.2.1.4: If $\Delta Q > 0$, accept the perturbation with probability $e^{-\Delta Q/T}$.

The performance of the Fourier space simulated annealing algorithm is illustrated in §5.4.2.

5.4.2 Computational Examples

Since Algorithm 5.2 operates in Fourier space, it is difficult to incorporate a positivity constraint into the algorithm. Consequently, the Fourier space simulated annealing algorithm is likely to perform most effectively, as compared to other phase retrieval algorithms, when recovering complex images. Results are now presented that are obtained when the Fourier space simulated annealing algorithm operates on the magnitude of the Fourier transform of the 8×8 pixel complex image shown in Figure 5.11.

(a) (b)

Figure 5.11 Original 8×8 pixel complex image; (a) magnitude and (b) phase.

The algorithm is begun with a pseudo-random starting phase for $\hat{\Phi}[p,q]$. The annealing schedule is chosen as follows. The value of the temperature T and the scale of perturbations α at the n_c^{th} cycle, denoted by T_{n_c} and α_{n_c} respectively, are computed by

$$T_{n_c} = c_1 T_{n_c-1} \tag{5.10a}$$

$$\alpha_{n_c} = c_2 (T_{n_c})^{c_3} \tag{5.10b}$$

where c_1, c_2, and c_3 are constants.

The first example comprises Algorithm 5.2 run with $N_s = 50$, $N_c = 50$, $T_0 = 0.1$, $c_1 = 0.8$, $c_2 = 50$, and $c_3 = 0.5$. Figure 5.12 shows the temperature T, and the cost function $Q\{\hat{\Phi}[p,q]\}$ (averaged over each scan) as the algorithm proceeds. Observe that once approximately 10 cycles have been performed, $Q\{\hat{\Phi}[p,q]\}$ begins to fall at the same rate as T. If $Q\{\hat{\Phi}[p,q]\}$ was to fall at a rate significantly different from that of T, this would indicate that the annealing schedule was inappropriately chosen. Note, however, there is always some value, determined by the level of contamination on the magnitude data, below which $Q\{\hat{\Phi}[p,q]\}$ can never fall.

It is informative to demonstrate the effect of choosing different parameters from those employed in the first example. Figure 5.12b shows the results obtained by running Algorithm 5.2 with $N_s = 20$, while the remaining parameters are identical to those employed in the first example. It is evident from Figure 5.12b that reducing the temperature too rapidly has resulted in the algorithm becoming trapped in a local minimum of the cost function $Q\{\hat{\Phi}[p,q]\}$. Figure 5.12c,d shows the results obtained by setting $c_2 = 25$ and $c_2 = 100$ respectively, while the remaining parameters are identical to those employed in the first example. For the case corresponding to Figure 5.12c, the perturbations are too small, which results in the algorithm becoming trapped in a local minimum of $Q\{\hat{\Phi}[p,q]\}$. Conversely, the perturbations are too large for the case corresponding to Figure 5.12d, and the algorithm once again becomes trapped.

It is now demonstrated that (5.10b), with $c_3 = 0.5$, is an effective means by which to vary α. To do so, it is necessary to examine the quantity p_{acc}, which is the ratio of the number of perturbations that are accepted to the total number of perturbations that are performed, in each scan. Figure 5.13a,b,c shows p_{acc} (more precisely, the values obtained by averaging p_{acc} over 100 scans) when Algorithm 5.2 operates with $c_3 = 0.25, 0.5, 1.0$ and $c_2 = 5, 25, 750$ respectively, while the remaining parameters in each case are $N_s = 1000, N_c = 30, T_0 = 0.1$, and $c_1 = 0.8$. The values of c_2 are chosen such that the value assumed by α when $T = 10^{-3}$ is approximately the same in all three cases. A sufficiently large N_s is chosen to ensure that the algorithm

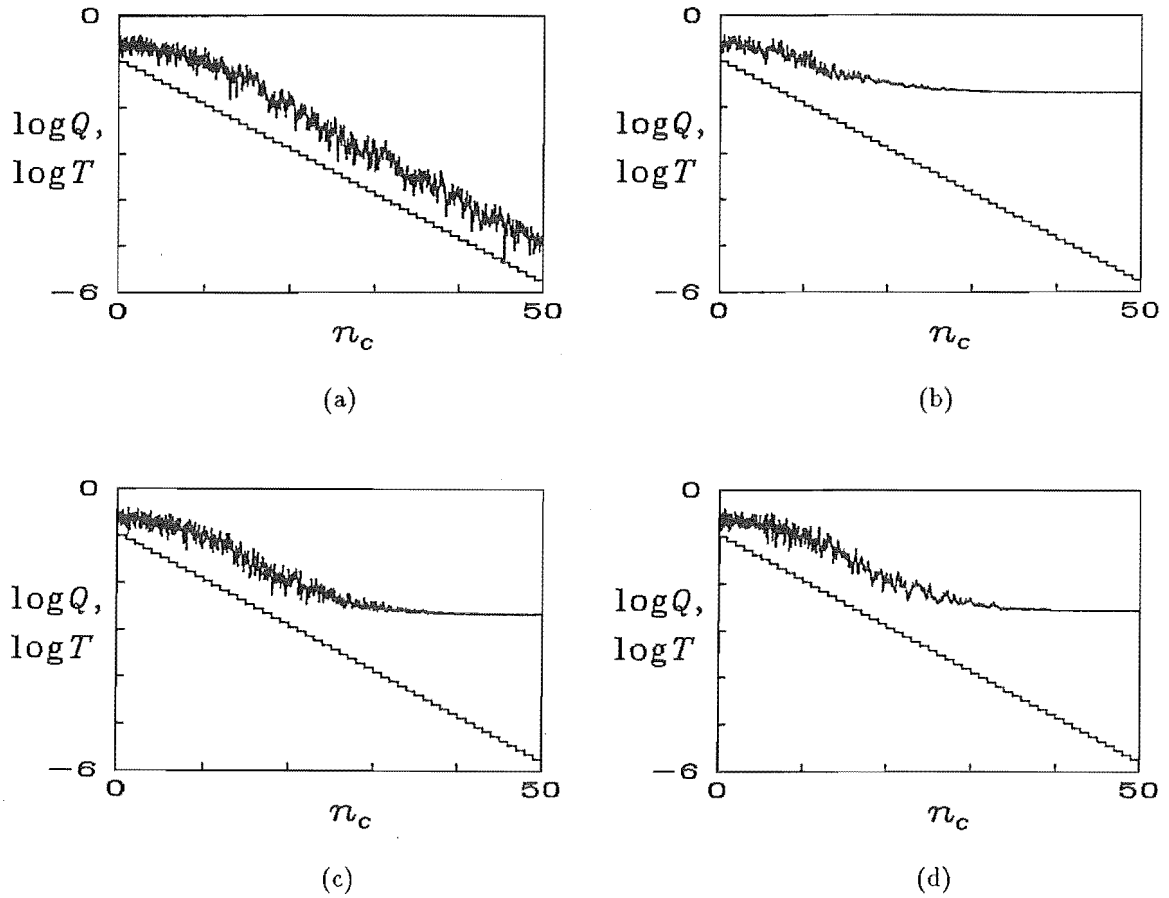


Figure 5.12 Temperature T (lower curve) and cost function $Q\{\hat{\Phi}[p,q]\}$ (upper curve) obtained when the Fourier space simulated annealing algorithm operates on the magnitude of the Fourier transform of the image shown in Figure 5.11, with (a) $N_s = 50$, $N_c = 50$, $T_0 = 0.1$, $c_1 = 0.8$, $c_2 = 50$, and $c_3 = 0.5$; (b) as for (a) but with $N_s = 20$; (c) as for (a) but with $c_2 = 25$; (d) as for (a) but with $c_2 = 100$.

converges in each case.

Figure 5.13b indicates that when $c_3 = 0.5$, roughly the same proportion of perturbations are accepted, after the first 10 or so cycles have been performed, despite the value of T being reduced over several orders of magnitude. However, it is evident from Figure 5.13a,c that when $c_3 = 0.25$ or $c_3 = 1.0$, the proportion of the perturbations that are accepted varies significantly as T is reduced. As mentioned in §3.1.2.2, the general simulated annealing algorithm tends to operate most efficiently when p_{acc} is neither too close to 0 nor too close to 1. Thus (5.10b) with $c_3 = 0.5$ appears to be an effective means by which to vary α.

The examples presented so far demonstrate that the choice of the annealing schedule is crucial to the effectiveness of the Fourier space simulated annealing algorithm. Further aspects of the algorithm are discussed in §5.4.3.

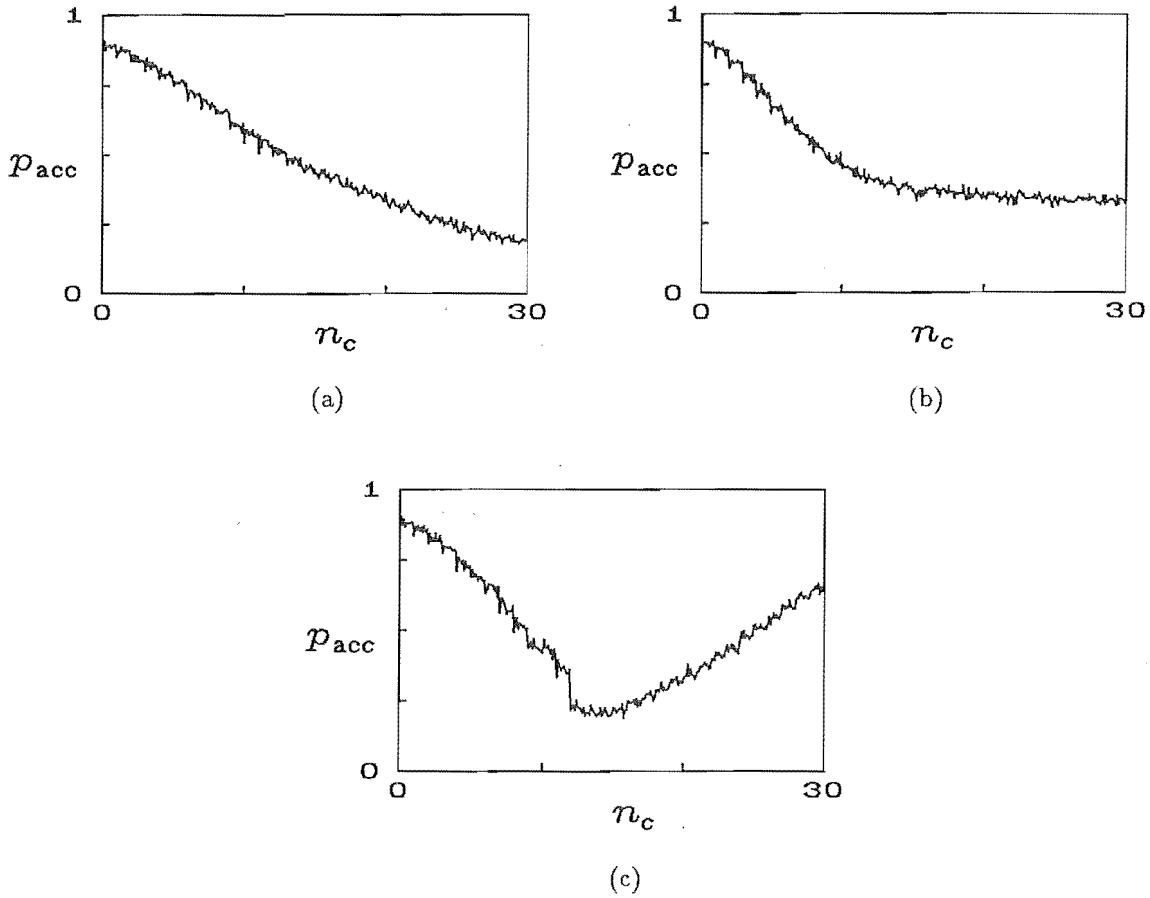


Figure 5.13 Fraction of perturbations that are accepted when the Fourier space simulated annealing algorithm operates with $N_s = 1000$, $N_c = 30$, $T_0 = 0.1$, $c_1 = 0.8$, and (a) $c_2 = 5$, $c_3 = 0.25$, (b) $c_2 = 25$, $c_3 = 0.5$, (c) $c_2 = 750$, $c_3 = 1.0$.

5.4.3 Discussion

The first example presented in §5.4.2 demonstrates that the Fourier space simulated annealing algorithm is indeed a feasible technique for effecting the phase retrieval of a complex image. The algorithm is also applicable to real images, in which case the odd symmetry of $\hat{\Phi}[p, q]$ should be exploited in step 1.2.1 of Algorithm 5.2. However, since this algorithm operates in Fourier space, it is not straightforward (although it is possible) to incorporate a positivity constraint into the algorithm. A positivity term could be included in the cost function, by employing one or more of the relationships described in §3.2.1 by which image positivity can be expressed as Fourier space inequalities. However, the computational effort involved in computing such a positivity term may well be significantly greater than that involved in computing the change in $Q\{\hat{\Phi}[p, q]\}$.

The Fourier space simulated annealing algorithm is undeniably computationally intensive. The first example presented in §5.4.2, in which an 8×8 pixel complex image is recovered, requires approximately 4 minutes of CPU time on a VAX-3500 computer (although the code was not written with efficiency particularly in mind).

Various enhancements are required before this technique could be viable for recovering images that are significantly more complicated than that shown in Figure 5.11. Some possible enhancements are now outlined.

Selection of an appropriate annealing schedule is crucial to the performance of the Fourier space simulated annealing algorithm. For the algorithm to be routinely applicable, strategies must be devised for automatically adjusting both T and α as the algorithm proceeds. T could well be varied in the manner described by Nieto-Vesperinas *et al.* (1988), whereby T is decreased only when the algorithm is deemed to have reached "equilibrium". Metropolis *et al.* (1953) and Corana *et al.* (1987) describe techniques for automatically adjusting α, which appear to be appropriate for this application. The basis of these methods is to adjust α so as to maintain p_{acc} (as defined in §5.4.2) at some preset level (typically 0.5).

Because of the computational expense of the Fourier space simulated annealing algorithm, it may be computationally advantageous to initially apply the algorithm to a reduced number of Fourier magnitude samples. This could be performed in a manner similar to that described in §5.2. Alternatively, appropriately combining the Fourier space simulated annealing algorithm with the hybrid input-output algorithm, in a similar manner to that in which Nieto-Vesperinas *et al.* (1988) combined their simulated annealing algorithm with the hybrid input-output algorithm, may prove to be an effective phase retrieval strategy.

It appears unlikely that the computational efficiency of the Fourier space simulated annealing algorithm could ever rival that of the hybrid input-output algorithm in situations where the hybrid input-output algorithm performs effectively. However, there are situations in which the hybrid input-output algorithm is known to perform poorly (Fienup, 1987; Lane, 1987). Such situations include the recovery of a complex image when only a loose support constraint is available, and the recovery of a complex image that possesses a support which is centro-symmetric (or nearly so). The Fourier space simulated annealing algorithm could well prove to be useful in such situations.

Chapter 6

Application of Simulated Annealing to the Blind Deconvolution Problem

The blind deconvolution problem is introduced in §2.6, and various methods for its solution are described in §3.3. A new method for solving the blind deconvolution problem, which appropriately invokes the simulated annealing algorithm (§3.1.2.2), is presented in this chapter. This method has already been reported by McCallum (1990).

The motivation for seeking new blind deconvolution methods is explained in §6.1, wherein the ill-posed nature of the blind deconvolution problem is discussed. The new simulated annealing based blind deconvolution algorithm is detailed in §6.2. Computational examples of the algorithm are presented in §6.3, with further aspects of the algorithm being discussed in §6.4.

6.1 Motivation

A number of blind deconvolution techniques are reviewed in §3.3. The zero-sheet based blind deconvolution method, as mentioned in §3.3.2, is presently ineffective when the given convolution data are contaminated. Of all the iterative techniques for blind deconvolution that are described in §3.3.3, the iterative blind deconvolution algorithm (§3.3.3.2) appears to be the most effective, and the most generally applicable.

Little appears to be known at present regarding the detailed behaviour of, and the limits to the performance of, the iterative blind deconvolution algorithm. It is not clear whether the iterative blind deconvolution algorithm will perform satisfactorily when operating on contaminated convolution data comprising a large number of pixels. It is thus worthwhile to seek alternative algorithms for effecting blind deconvolution. After all, it is always desirable to have more than one technique available for solving an image reconstruction problem. This then allows the measured data to be processed by different algorithms, so that the resulting reconstructed images can be compared.

When attempting to solve a blind deconvolution problem, one ought to bear in

mind that all blind deconvolution problems are inherently ill-conditioned. This point is now explained in greater detail. Consider a discrete two-dimensional convolution $g[m,n]$, given by

$$g[m,n] = f[m,n] \odot h[m,n] + c[m,n] \tag{6.1}$$

where $f[m,n]$ and $h[m,n]$ are the components of the convolution, and $c[m,n]$ is the contamination. Recall, from §2.6, that solving the blind deconvolution problem amounts to finding estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ of $f[m,n]$ and $h[m,n]$ respectively, such that $\hat{f}[m,n] \odot \hat{h}[m,n]$ is "close" to $g[m,n]$. In fact, the best that one can hope to do is to find some $\hat{f}[m,n]$ and $\hat{h}[m,n]$ such that $g[m,n] - \hat{f}[m,n] \odot \hat{h}[m,n]$ is statistically similar to $c[m,n]$. It is now convenient to express $\hat{f}[m,n]$ and $\hat{h}[m,n]$ as

$$\hat{f}[m,n] = f[m,n] + e_f[m,n] \quad \text{and} \quad \hat{h}[m,n] = h[m,n] + e_h[m,n] \tag{6.2}$$

where $e_f[m,n]$ and $e_h[m,n]$ are the *reconstruction errors* of $\hat{f}[m,n]$ and $\hat{h}[m,n]$ respectively. The blind deconvolution problem is said to be *ill-posed* because a small $c[m,n]$ (as compared to $g[m,n]$) tends to give rise to a larger $e_f[m,n]$ (as compared to $f[m,n]$) and/or a larger $e_h[m,n]$ (as compared to $h[m,n]$).

An example is now presented which illustrates the ill-posed nature of the blind deconvolution problem. Two 8×8 pixel images $f[m,n]$ and $h[m,n]$ are shown in Figure 6.1a,b, with their convolution being shown in Figure 6.1c. Estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ are computed via (6.2), in which $e_f[m,n]$ and $e_h[m,n]$ are generated pseudo-randomly. The pixels of $e_f[m,n]$ and $e_h[m,n]$ are uniformly distributed with zero-mean such that $\mathrm{En}\{e_f\}/\mathrm{En}\{f\} = \mathrm{En}\{e_h\}/\mathrm{En}\{h\} = 0.1$. Figure 6.1d–f shows $\hat{f}[m,n]$, $\hat{h}[m,n]$ and $\hat{f}[m,n] \odot \hat{h}[m,n]$ respectively. Note particularly that the degree by which $\hat{f}[m,n] \odot \hat{h}[m,n]$ differs from $f[m,n] \odot h[m,n]$ is significantly less than that by which $\hat{f}[m,n]$ differs from $f[m,n]$ and $\hat{h}[m,n]$ differs from $h[m,n]$. This observation is confirmed by noting that $\mathrm{En}\{\hat{f} \odot \hat{h} - f \odot h\}/\mathrm{En}\{f \odot h\} = 0.015$. This value is significantly less than that of $\mathrm{En}\{e_f\}/\mathrm{En}\{f\}$ and $\mathrm{En}\{e_h\}/\mathrm{En}\{h\}$, both of which are 0.1.

The reason for $e_f[m,n]$ and $e_h[m,n]$ being more apparent in $\hat{f}[m,n]$ and $\hat{h}[m,n]$ than in $\hat{f}[m,n] \odot \hat{h}[m,n]$ is clarified by expressing $\hat{f}[m,n] \odot \hat{h}[m,n]$ as

$$\begin{aligned} \hat{f}[m,n] \odot \hat{h}[m,n] &= f[m,n] \odot h[m,n] + f[m,n] \odot e_h[m,n] \\ &+ h[m,n] \odot e_f[m,n] + e_f[m,n] \odot e_h[m,n] \end{aligned} \tag{6.3}$$

which follows from (6.2). If $\hat{f}[m,n]$ and $\hat{h}[m,n]$ are to serve as useful estimates of $f[m,n]$ and $h[m,n]$, then $e_f[m,n]$ and $e_h[m,n]$ must be significantly smaller than $f[m,n]$ and $h[m,n]$ respectively. Thus the fourth term in (6.3) must be significantly smaller than the preceding two terms. Now the second term in (6.3) is effectively a smoothed version of $e_h[m,n]$ (the degree of smoothing depending on the form of $f[m,n]$), while the third term is a smoothed version of $e_f[m,n]$. Thus the contributions to $\hat{f}[m,n] \odot \hat{h}[m,n]$ due to $e_f[m,n]$ and $e_h[m,n]$ are expected to be significantly less than the contributions to $\hat{f}[m,n]$ and $\hat{h}[m,n]$ due to $e_f[m,n]$ and $e_h[m,n]$ respectively.

The significance to the blind deconvolution problem of the example presented in Figure 6.1 is now explained. It is first necessary to define the *contamination level*

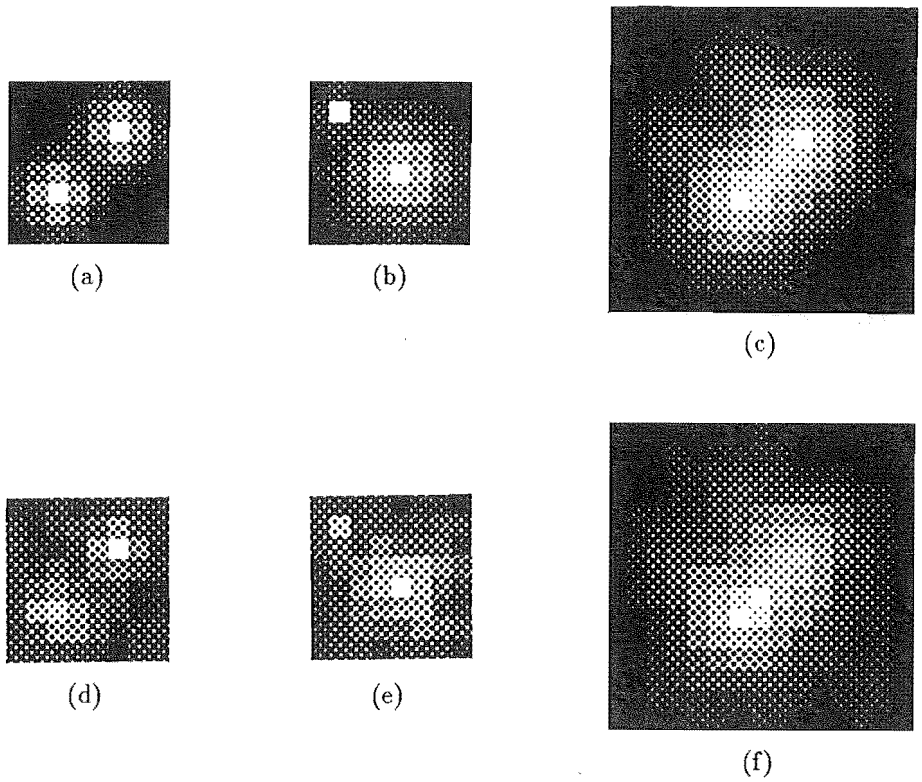


Figure 6.1 Illustration of the ill-posed nature of the blind deconvolution problem. (a,b) original components $f[m,n]$ and $h[m,n]$, (c) the convolution $f[m,n] \odot h[m,n]$, (d,e) image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$, and (f) the convolution $\hat{f}[m,n] \odot \hat{h}[m,n]$.

of $g[m,n]$, denoted by cl, as

$$cl = \mathrm{En}\{c[m,n]\} \,/\mathrm{En}\{g[m,n]\} \tag{6.4}$$

If presented with the contaminated convolution $g[m,n]$ of the components $f[m,n]$ and $h[m,n]$ depicted in Figure 6.1a,b, for which it is estimated that $cl = 0.015$, then those $\hat{f}[m,n]$ and $\hat{h}[m,n]$ shown in Figure 6.1d,e would have to be accepted as being valid solutions. So too would a multitude of other solutions, which differ from $f[m,n]$ and $h[m,n]$ in a similar manner to that in which those $\hat{f}[m,n]$ and $\hat{h}[m,n]$ shown in Figure 6.1d and e differ from the $f[m,n]$ and $h[m,n]$ shown in Figure 6.1a and b respectively.

It is worth pointing out that, when solving blind deconvolution problems, it is often possible to obtain solutions $\hat{f}[m,n]$ and $\hat{h}[m,n]$ for which the major proportion of the energy of $e_f[m,n]$ and/or $e_h[m,n]$ exists at high spatial frequencies. This can occur if the energy of $h[m,n]$ and/or $f[m,n]$ is concentrated at low spatial frequencies (as is often so in practice). The combination of $f[m,n]$ being low-pass with $e_h[m,n]$ being high-pass then results in the energy of the second term in (6.2) being spread approximately uniformly throughout Fourier space. The third term in (6.2) is similarly affected if $h[m,n]$ is low-pass and $e_f[m,n]$ is high-pass.

The preceding discussion has attempted to explain why blind deconvolution is an ill-posed problem. It is advantageous to employ regularization methods, as men-

tioned in §3.1.2.3, when attempting to solve ill-posed problems. Evidently, any blind deconvolution algorithm into which some form of regularization could be incorporated would be highly desirable.

A new method for solving the blind deconvolution problem, by appropriately invoking the simulated annealing algorithm, is described in this chapter. The algorithm is detailed in §6.2, and computational examples are presented in §6.3. Further aspects of the algorithm, including the possibility of incorporating a regularization scheme, are discussed in §6.4.

6.2 The Simulated Annealing Blind Deconvolution Algorithm

The blind deconvolution problem is posed in §2.6. Furthermore, §3.1.2 describes how a general image reconstruction problem may be posed as a constrained minimization problem. The blind deconvolution problem, in which two-dimensional discrete images are assumed, is now posed as a constrained minimization problem, following the technique described in §3.1.2, as:

> "Find estimates of $f[m,n]$ and $h[m,n]$, denoted by $\hat{f}[m,n]$ and $\hat{h}[m,n]$ respectively, which minimize the cost function $Q\{\hat{f},\hat{h}\}$, as defined by
>
> $$Q\{\hat{f},\hat{h}\} = \mathrm{En}\{\hat{f} \odot \hat{h} - g\}/\mathrm{En}\{g\}, \tag{6.5}$$
>
> subject to any given constraints upon $\hat{f}[m,n]$ and $\hat{h}[m,n]$."

Recall from §2.6 that the Fourier phase problem is equivalent to a special case of the blind deconvolution problem, in which $h[m,n] = f^*[-m,-n]$. In this case, with $\hat{h}[m,n] = (\hat{f}[-m,-n])^*$, the cost function $Q\{\hat{f},\hat{h}\}$ is, apart from a scale factor, equivalent to the cost function $Q'\{\hat{f}[m,n]\}$ (as defined by (3.69)) associated with the Fourier phase problem. As mentioned in §3.2.9.1, Nieto-Vesperinas (1986) has demonstrated that the cost function $Q'\{\hat{f}[m,n]\}$ is multimodal. It can thus be concluded that $Q\{\hat{f},\hat{h}\}$ must also, in general, be multimodal.

The simulated annealing algorithm, as described in §3.1.2.2, appears to be the most generally applicable technique that is currently available for minimizing multimodal functions. The new blind deconvolution method that is about to be described appropriately invokes the simulated annealing algorithm to minimize $Q\{\hat{f},\hat{h}\}$. The resulting algorithm is here called the *simulated annealing blind deconvolution algorithm.*

The simulated annealing algorithm is here applied to the problem of minimizing $Q\{\hat{f},\hat{h}\}$, in order to effect blind deconvolution, in a similar manner to that in which Nieto-Vesperinas *et al.* (1988) apply it to the problem of minimizing $Q'\{\hat{f}[m,n]\}$, in order to effect phase retrieval. The general simulated annealing algorithm is described in detail in §3.1.2.2, while the method by which Nieto-Vesperinas *et al.* (1988) apply the simulated annealing algorithm to the Fourier phase problem is described in §3.2.9.2. Both positivity and support constraints are enforced by the algorithm described here, although it is possible (but less efficient) to enforce only one of these constraints.

In order to apply a support constraint to $\hat{f}[m,n]$, it is necessary to define the mask $M_f[m,n]$, the pixels of which are set to unity within the estimated support of $f[m,n]$, and are set to zero elsewhere. A mask $M_h[m,n]$ is similarly defined to enforce a support constraint upon $\hat{h}[m,n]$. Note that the position of $M_f[m,n]$ with respect to $M_h[m,n]$ should be chosen with care, as mentioned in §3.3.3.2, to ensure that $g[m,n]$ lies within the non-zero part of $M_f[m,n] \odot M_h[m,n]$.

The support constraint is enforced upon $\hat{f}[m,n]$ ($\hat{h}[m,n]$) by only perturbing those pixels of $\hat{f}[m,n]$ ($\hat{h}[m,n]$) at which $M_f[m,n] = 1$ ($M_h[m,n] = 1$). The pixels of $\hat{f}[m,n]$ ($\hat{h}[m,n]$) at which $M_f[m,n] = 0$ ($M_h[m,n] = 0$) are initially set to zero, and remain so while the algorithm proceeds. Hence $\hat{f}[m,n] \odot \hat{h}[m,n]$ can only ever assume non-zero values at those pixels for which $M_f[m,n] \odot M_h[m,n]$ is non-zero. Thus $Q\{\hat{f},\hat{h}\}$ need only be computed at those pixels for which $M_f[m,n] \odot M_h[m,n]$ is non-zero. A positivity constraint is enforced upon $\hat{f}[m,n]$ and $\hat{h}[m,n]$ by simply setting to zero any pixel of $\hat{f}[m,n]$ or $\hat{h}[m,n]$ which assumes a negative value following a perturbation.

Minimizing $Q\{\hat{f},\hat{h}\}$ to solve the blind deconvolution problem evidently involves estimating two images, whereas minimizing $Q'\{\hat{f}[m,n]\}$ to solve the phase problem only requires the estimation of a single image. One scan of the simulated annealing blind deconvolution algorithm thus consists of sequentially perturbing each pixel of $\hat{f}[m,n]$ at which $M_f[m,n] = 1$, then sequentially perturbing each pixel of $\hat{h}[m,n]$ at which $M_h[m,n] = 1$.

In the absence of any better starting estimates, the algorithm is begun with pseudo-random images $\hat{f}_0[m,n]$ and $\hat{h}_0[m,n]$ as estimates of $f[m,n]$ and $h[m,n]$ respectively. The pixels of $\hat{f}_0[m,n]$ and $\hat{h}_0[m,n]$ at which $M_f[m,n] = 1$ and $M_h[m,n] = 1$ respectively are uniformly distributed in the range $[0,\delta]$, where δ is chosen such that the rms value of the aforesaid pixels is unity. To ensure that the algorithm functions in an identical manner when operating upon scaled versions of the same convolution data, it is convenient to initially scale $g[m,n]$ such that $\mathrm{En}\{g\} = \mathrm{En}\{\hat{f}_0 \odot \hat{h}_0\}$ following such scaling.

The simulated annealing blind deconvolution algorithm comprises the following steps:

Algorithm 6.1

Step 1: Set cycle counter $n_c = 1$, and set scan counter $n_s = 1$.

Step 2: Calculate values for T (temperature) and α (scale of perturbations).

Step 3: Multiply $\hat{f}[m,n]$ and $\hat{h}[m,n]$ by factors β and $1/\beta$ respectively, where β is chosen such that $\mathrm{rms}\{\hat{f}\} = \mathrm{rms}\{\hat{h}\}$ following the scaling operation.

Step 4: Repeat the following 4 steps for each pixel $[m',n']$ for which $M_f[m',n'] = 1$:

Step 4.1: Form the perturbed image $\hat{f}_p[m,n]$ as

$$\hat{f}_p[m,n] = \begin{cases} \hat{f}[m,n] + \alpha\, r_1 & \text{if } [m,n] = [m',n'] \\ \hat{f}[m,n] & \text{otherwise} \end{cases} \tag{6.6}$$

where r_1 is a pseudo-random number, uniformly distributed in the range $[-0.5, 0.5]$.

Step 4.2: If $\hat{f}_p[m', n'] < 0$, then set $\hat{f}_p[m', n'] = 0$.

Step 4.3: Calculate $\Delta Q = Q(\hat{f}_p, \hat{h}) - Q(\hat{f}, \hat{h})$.

Step 4.4: If $\Delta Q \leq 0$, or if $\Delta Q > 0$ and $e^{-\Delta Q/T} < r_2$ (where r_2 is a pseudo-random number, uniformly distributed in the range $[0, 1]$), then accept the perturbation (i.e. set $\hat{f}[m', n'] = \hat{f}_p[m', n']$), otherwise reject the perturbation (i.e. make no change to $\hat{f}[m', n']$).

Step 5: Repeat Step 4, interchanging $\hat{f}[m, n]$, $\hat{f}_p[m, n]$ and $M_f[m, n]$ for $\hat{h}[m, n]$, $\hat{h}_p[m, n]$ and $M_h[m, n]$ respectively. Note that step 4 together with step 5 constitutes one scan.

Step 6: If $n_s < N_s$, then increment n_s and return to step 4. Note that a sequence of N_s scans constitutes one cycle.

Step 7: If $n_c < N_c$, then increment n_c and return to step 2, otherwise stop.

Some aspects of Algorithm 6.1 require further explanation. The manner in which α and T are varied in step 2 (i.e. the annealing schedule) is crucial, as explained in §3.1.2.2, to ensuring that the algorithm converges to the global minimum, or an acceptable local minimum, of $Q\{\hat{f}, \hat{h}\}$. Suitable annealing schedules are described in §6.3.

Step 3 is required to avoid differences developing between the rms values of $\hat{f}[m, n]$ and $\hat{h}[m, n]$. The perturbations are made to $\hat{f}[m, n]$ and $\hat{h}[m, n]$ with the same value of α. It is thus desirable for $\hat{f}[m, n]$ and $\hat{h}[m, n]$ to have similar rms values, to ensure that the relative size of the perturbations made to $\hat{f}[m, n]$ and $\hat{h}[m, n]$ are similar.

Note that step 4.3 can be performed efficiently, provided that the convolution $\hat{f}[m, n] \odot \hat{h}[m, n]$ is saved following step 4.4. The new convolution $\hat{f}_p[m, n] \odot \hat{h}[m, n]$ can then be computed, when evaluating $Q(\hat{f}_p, \hat{h})$ at step 4.3, by

$$\hat{f}_p[m, n] \odot \hat{h}[m, n] = \hat{f}[m, n] \odot \hat{h}[m, n] + p\,\hat{h}[m - m', n - n'] \tag{6.7}$$

where p is the perturbation, given by $p = \hat{f}_p[m', n'] - \hat{f}[m', n']$. Furthermore, ΔQ need only be evaluated over the region in which $M_f[m - m', n - n'] = 1$, since $\hat{f}[m, n] \odot \hat{h}[m, n]$ is identical to $\hat{f}_p[m, n] \odot \hat{h}[m, n]$ outside this region.

Computational examples of the simulated annealing blind deconvolution algorithm are presented in §6.3.

6.3 Computational Examples

This section presents computational examples that demonstrate various aspects of the performance of the simulated annealing blind deconvolution algorithm. An annealing schedule that has been found to perform well, and which is similar to that employed in

§5.4.2, is now described. The value of the temperature T and the scale of perturbations α at the n_c^{th} cycle, denoted by T_{n_c} and α_{n_c} respectively, are computed from

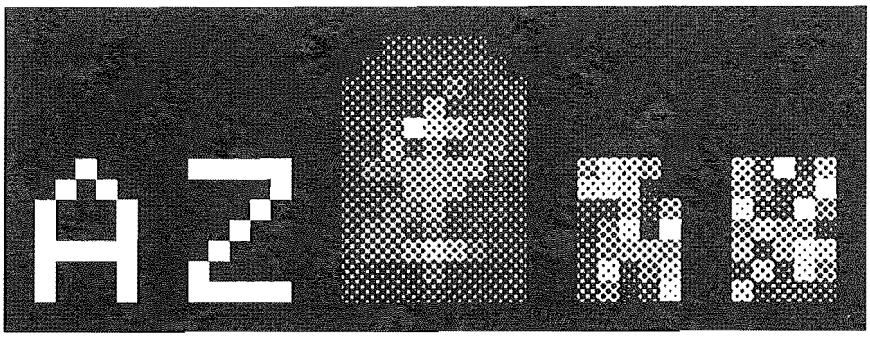
$$T_1 = c_1 Q_0 \tag{6.8a}$$

$$T_{n_c} = c_2 T_{n_c-1} \tag{6.8b}$$

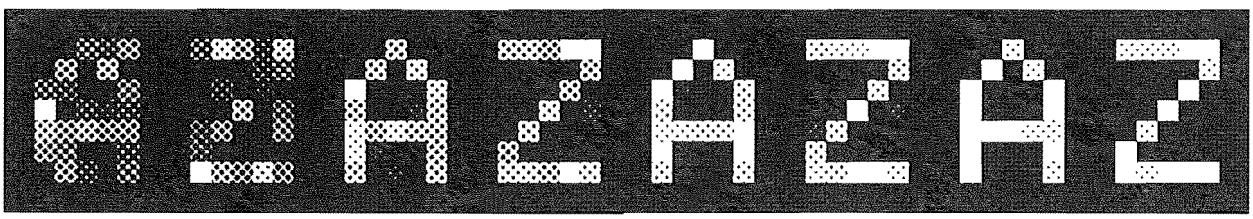
$$\alpha_{n_c} = c_3 (T_{n_c})^{c_4} \tag{6.8c}$$

where c_1, c_2, c_3, and c_4 are constants, and Q_0 is the initial cost, given by $Q_0 = Q\{\hat{f}_0[m,n], \hat{h}_0[m,n]\}$.

The first example comprises the simulated annealing blind deconvolution algorithm applied to the 9×13 pixel convolution, presented in Figure 6.2c, of the two 5×7 pixel components shown in Figure 6.2a,b. A mask which is nonzero within a 5×7 pixel rectangle is chosen for both $M_f[m,n]$ and $M_h[m,n]$. The algorithm is begun with the starting images shown in Figure 6.2d,e, for which $Q_0 = 0.20$. The values employed for the various parameters are $N_s = 50$, $N_c = 40$, $c_1 = 0.1$, $c_2 = 0.8$, $c_3 = 100$, and $c_4 = 0.5$. The resulting image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ obtained when $n_c = 10$, 20, 30, and 40 are shown in Figure 6.2f–m, for which the corresponding values of $Q\{\hat{f}, \hat{h}\}$ are 7.7×10^{-2}, 5.2×10^{-3}, 5.3×10^{-4}, and 6.4×10^{-5}. Note the improvement in the faithfulness of $\hat{f}[m,n]$ and $\hat{h}[m,n]$ as n_c increases. Figure 6.3a shows the temperature T, and the cost function $Q\{\hat{f}, \hat{h}\}$ (averaged over each scan), obtained as the algorithm proceeds. Observe that once a few cycles have been completed, $Q\{\hat{f}, \hat{h}\}$ begins to fall at the same rate as T. If $Q\{\hat{f}, \hat{h}\}$ should fall at a rate



(a) (b) (c) (d) (e)



(f) (g) (h) (i) (j) (k) (l) (m)

Figure 6.2 Demonstration of the simulated annealing blind deconvolution algorithm, with $N_s = 50$, $N_c = 40$, $c_1 = 0.1$, $c_2 = 0.8$, $c_3 = 100$, and $c_4 = 0.5$. (a,b) original images $f[m,n]$ and $h[m,n]$, (c) the convolution $g[m,n] = f[m,n] \odot h[m,n]$, (d,e) starting images $\hat{f}_0[m,n]$ and $\hat{h}_0[m,n]$, and image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ obtained after (f,g) 10, (h,i) 20, (j,k) 30, and (l,m) 40 cycles have been performed.

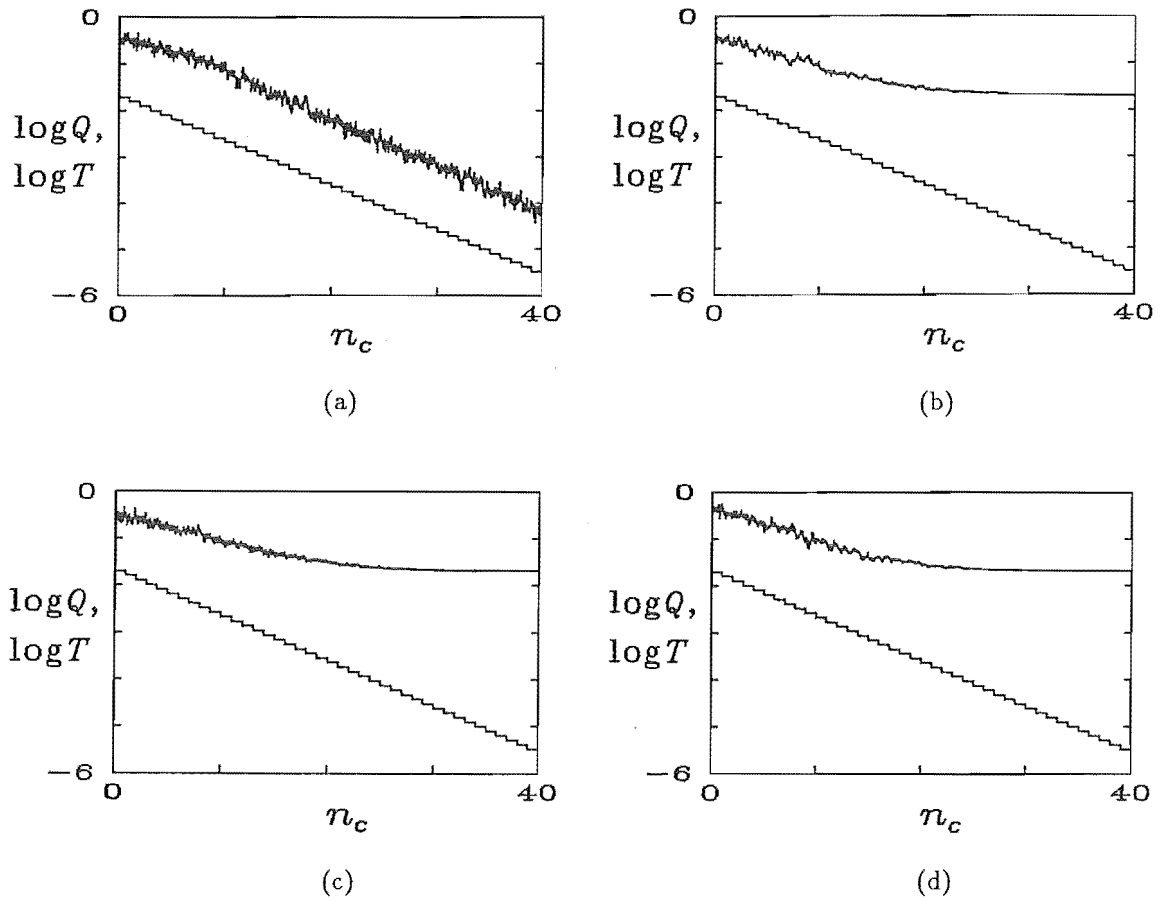


Figure 6.3 Cost function $Q\{\hat{f}, \hat{h}\}$ (upper curve) and temperature T (lower curve) which result when the simulated annealing blind deconvolution algorithm operates on the convolution presented in Figure 6.2c with (a) $N_s = 50$, $N_c = 40$, $c_1 = 0.1$, $c_2 = 0.8$, $c_3 = 100$, and $c_4 = 0.5$, (b) as for (a) but with $N_s = 15$, (c) as for (a) but with $c_3 = 30$, (d) as for (a) but with $c_3 = 300$.

significantly different from that of T, it is then clear that the annealing schedule is inappropriately chosen.

The effect on the algorithm's behaviour of carelessly choosing the annealing schedule is now demonstrated. Figure 6.3b–d shows the results obtained when operating the algorithm with $N_s = 15$, $c_3 = 30$, and $c_3 = 300$ respectively, while the remaining parameters in each case are identical to those employed in the first example. In all three cases, the algorithm becomes trapped in a local minimum of the cost function $Q\{\hat{f}, \hat{h}\}$. This is due, respectively, to the temperature being lowered too rapidly, the perturbations being too small, and the perturbations being too large.

It is now demonstrated that (6.8c), with $c_4 = 0.5$, is an effective means by which to vary α. It has already been shown, in §5.4.2, that this is so in the case of the Fourier space simulated annealing algorithm for phase retrieval. However, since the cost function employed here is different from that in §5.4.2, it is necessary to also show that it is so in this case. The quantity p_{acc} is now defined as the ratio of the number of perturbations that are accepted to the total number of perturbations that are performed, in each cycle, excluding those perturbations at which $\Delta Q = 0$ (as

occurs if a zero valued pixel is perturbed by a negative amount, when a positivity constraint is enforced).

Figure 6.4a,b,c shows the variation of p_{acc} when the algorithm operates with $c_4 = 0.25, 0.5, 1.0$ and $c_3 = 13, 100, 6000$ respectively, with $N_s = 200$, while the remaining parameters are those employed in the first example. It is evident that p_{acc} is maintained at an approximately constant value, despite T being reduced by approximately four orders of magnitude, when $c_4 = 0.5$. Recall from §3.1.2.2 that the general simulated annealing algorithm tends to operate most efficiently when p_{acc} is neither too close to 0 nor too close to 1. The results presented in Figure 6.4 thus indicate that (6.8c) with $c_4 = 0.5$ is indeed an effective means by which to vary α, provided that c_3 is chosen with care.

An alternative means by which α may be varied, which eliminates the need to choose a value for c_3, involves automatically adjusting α so as to maintain an acceptable value of p_{acc}. Both Metropolis *et al.* (1953) and Corana *et al.* (1987) describe such techniques. An appropriate strategy, which is straightforward to implement, is to adjust α by

$$\alpha_{n_c+1} = \begin{cases} 0.8\,\alpha_{n_c} & \text{if } p_{\text{acc},n_c} < 0.4 \\ \alpha_{n_c} & \text{if } 0.4 \leq p_{\text{acc},n_c} < 0.6 \\ \alpha_{n_c}/0.8 & \text{if } p_{\text{acc},n_c} \geq 0.6 \end{cases} \tag{6.9}$$

in step 2 of Algorithm 6.1, where p_{acc,n_c} is the value of p_{acc} computed at the n_c^{th} cycle. The algorithm is begun with $\alpha_1 = 100\sqrt{T_1}$, and the value of T is initially held constant until p_{acc} assumes a value in the range $[0.4, 0.6]$. Figure 6.5 shows the values of α, $Q\{\hat{f}, \hat{h}\}$, and T which result when the algorithm operates with α being varied by the abovementioned method, but with the remaining parameters identical to those employed in the first example. Note particularly that α falls at approximately the same rate as $\sqrt{T}$.

The performance of the algorithm is now demonstrated when quenching (i.e. running the algorithm with $T = 0$, as described in §3.1.2.2) rather than annealing is employed. The algorithm is run with $T = 0$ and $\alpha = 1.0$, while the remaining parameters are those employed in the first example. The results obtained when two different pairs of starting images are employed are shown in Figure 6.6, with the corresponding image estimates being shown in Figure 6.7. Whilst the first case stagnates at a high value of $Q\{\hat{f}, \hat{h}\}$, for which the reconstructed images are unrecognizable, the second case stagnates at a much lower value of $Q\{\hat{f}, \hat{h}\}$, for which the reconstructed images are clearly recognizable.

The performance of the simulated annealing blind deconvolution algorithm, when operating on contaminated convolution data, is now demonstrated. The algorithm is applied to the contaminated ($cl = 10^{-3}$ and 10^{-2}) versions of the convolution presented in Figure 6.2c, with the same parameters as those employed in the first example. Figure 6.8 shows the resulting behaviour of $Q\{\hat{f}, \hat{h}\}$, with the corresponding image estimates $\hat{f}[m, n]$ and $\hat{h}[m, n]$ being presented in Figure 6.9. Note that, in both cases, $Q\{\hat{f}, \hat{h}\}$ initially declines at a similar rate to T, before levelling off at a value slightly below the contamination level.

As a final example, the simulated annealing blind deconvolution algorithm is applied to the 24×23 pixel convolution, shown in Figure 6.10c, of the 17×16

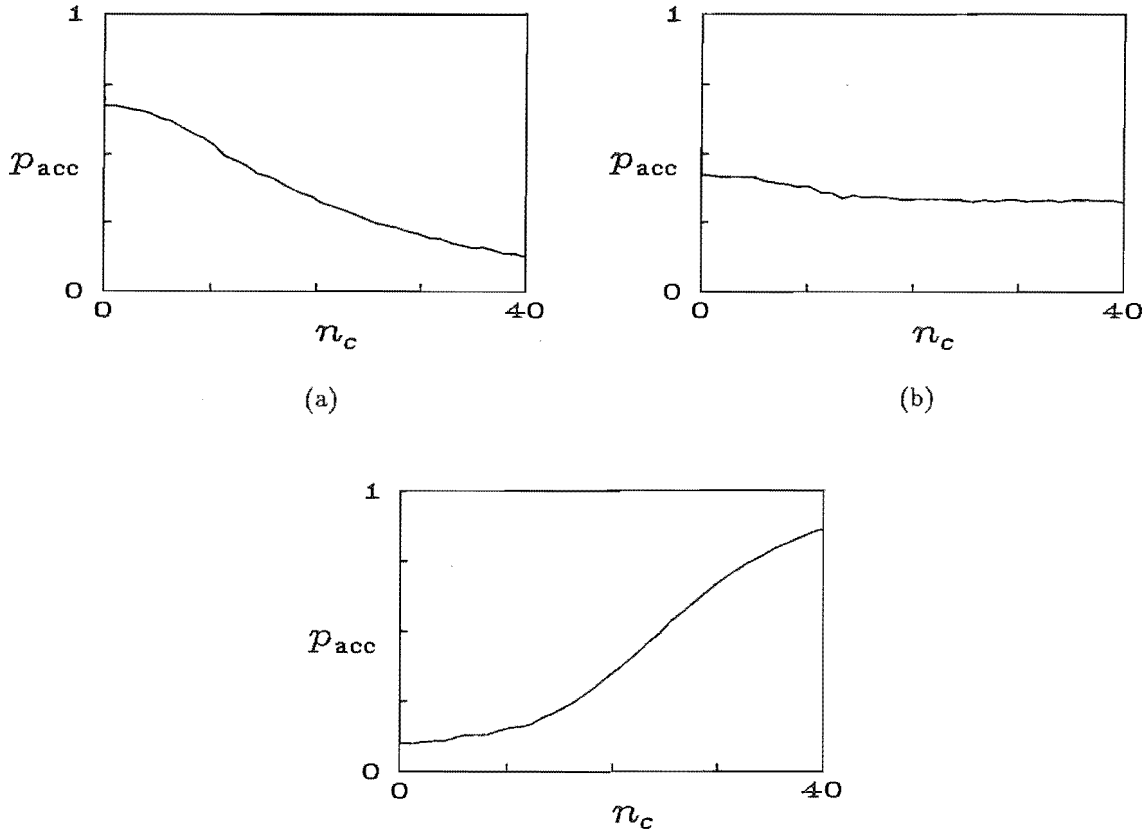


Figure 6.4 The fraction of perturbations that are accepted, p_{acc}, when the simulated annealing blind deconvolution algorithm operates with $N_s = 200$, $N_c = 40$, $c_1 = 0.1$, $c_2 = 0.8$, and (a) $c_4 = 0.25$, $c_3 = 13$, (b) $c_4 = 0.5$, $c_3 = 100$, (c) $c_4 = 1.0$, $c_3 = 6000$.

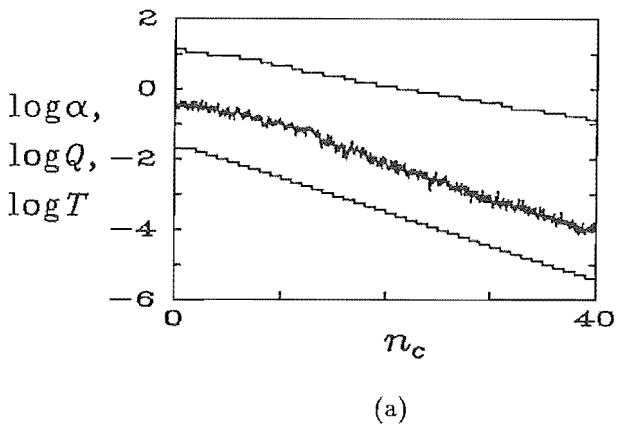


Figure 6.5 Results obtained when the automatic α adjustment strategy, as specified by (6.9), is incorporated into the simulated annealing blind deconvolution algorithm, with $N_s = 50$, $N_c = 40$, $c_1 = 0.1$, and $c_2 = 0.8$. The upper, middle, and lower curves are α, Q, and T respectively.

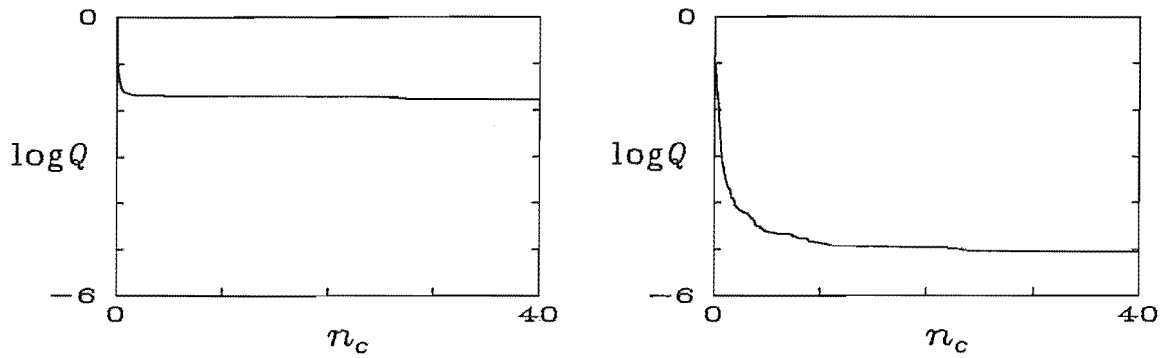


Figure 6.6 Behaviour of the cost function when quenching, with $N_s = 50$, $N_c = 40$, $T = 0$, and $\alpha = 1$, for two different pairs of starting images.

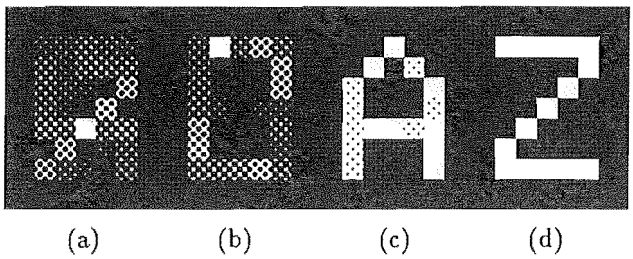


Figure 6.7 Image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ corresponding to (a,b) Figure 6.6a and (c,d) Figure 6.6b.

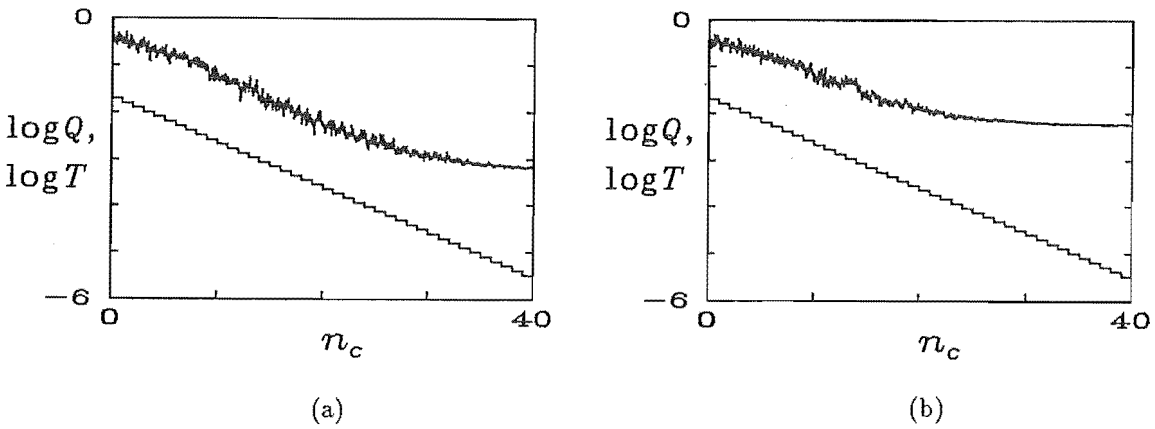
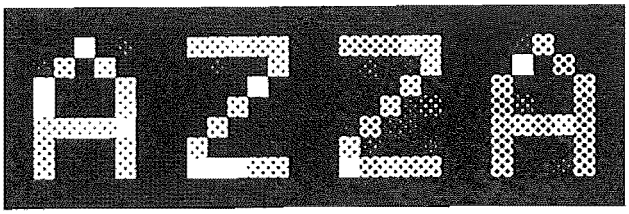
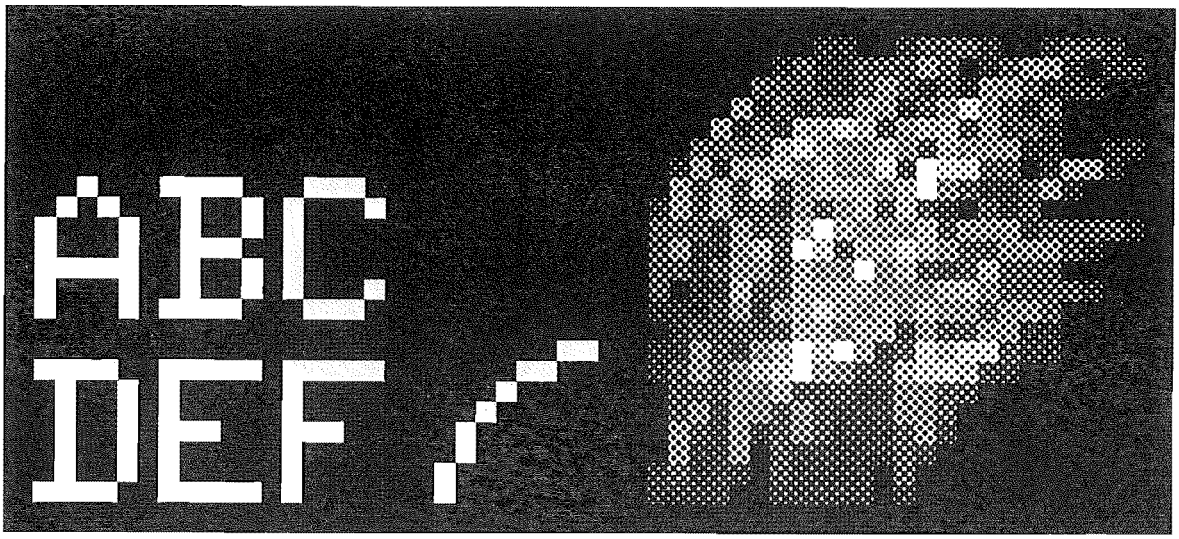


Figure 6.8 Behaviour of the cost function when the simulated annealing blind deconvolution algorithm operates on the contaminated convolution of Figure 6.2c, with the parameters used to generate Figure 6.3a, for (a) $cl = 10^{-3}$ and (b) $cl = 10^{-2}$. The upper and lower curves are Q and T respectively.

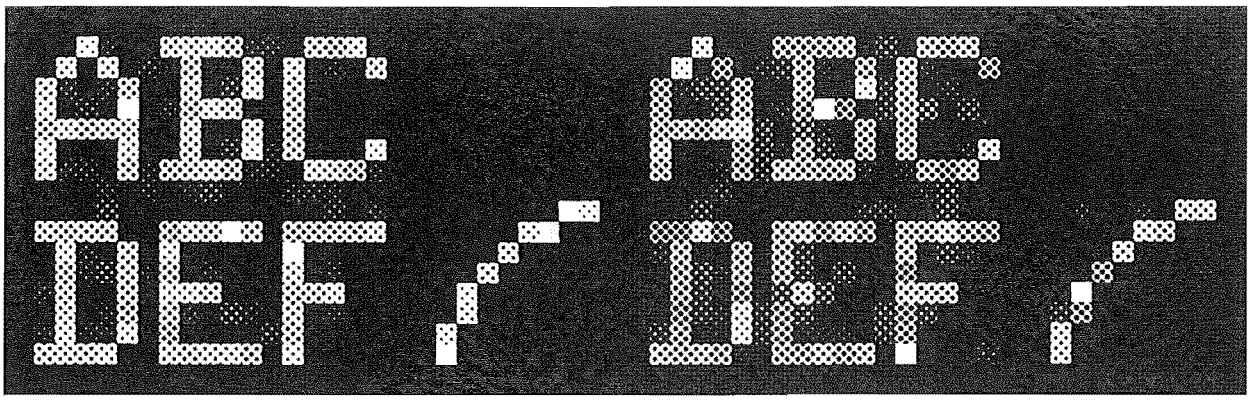


(a) (b) (c) (d)

Figure 6.9 Image estimates corresponding to (a,b) Figure 6.8a and (c,d) Figure 6.8b.



(a) (b) (c)



(d) (e) (f) (g)

Figure 6.10 Second example of the simulated annealing blind deconvolution algorithm. (a,b) original images $f[m,n]$ and $h[m,n]$, (c) the convolution $g[m,n] = f[m,n] \odot h[m,n]$, and image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ for (d,e) $cl = 3.3 \times 10^{-3}$ and (f,g) $cl = 3.3 \times 10^{-2}$.

and 8×8 pixel components shown in Figure 6.10a,b. The algorithm is run with $N_s = 25$, $N_c = 30$, $c_1 = 0.01$, and $c_2 = 0.8$, with α being adjusted automatically in the previously described manner. Image estimates $\hat{f}[m,n]$ and $\hat{h}[m,n]$ obtained when $cl = 3.3 \times 10^{-3}$ and $cl = 3.3 \times 10^{-2}$ are shown in Figure 6.10d–g, for which $Q\{\hat{f},\hat{h}\} = 2.8 \times 10^{-3}$ and 2.3×10^{-2} respectively. The simulated annealing blind deconvolution algorithm has thus, in both cases, succeeded in finding a solution that is consistent with the given convolution data, to within the uncertainty in the measured data.

The results of the simulated annealing blind deconvolution algorithm presented in this section are discussed further in §6.4.

6.4 Discussion

The examples presented in §6.3 clearly demonstrate that the simulated annealing blind deconvolution algorithm is a viable technique for effecting the blind deconvolution of contaminated convolution data. However, it is evident from the four examples shown in Figure 6.3 that the choice of the annealing schedule is crucial to the effectiveness of the algorithm.

The examples shown in Figure 6.4 indicate that an effective means of varying α is by setting $\alpha_{n_c} = c_3\sqrt{T_{n_c}}$. Since $Q\{\hat{f},\hat{h}\}$ falls at the same rate as T when the algorithm is operating correctly, the algorithm should behave similarly if α is varied by setting $\alpha_{n_c} = c_3'\sqrt{Q_{n_c}}$ (where Q_{n_c} is the value of $Q\{\hat{f},\hat{h}\}$ averaged over the n_c^{th} cycle), provided that c_3' is appropriately selected. It is worth mentioning that varying α in either of the above manners should also be effective in the simulated annealing algorithm for phase retrieval, because of its close relationship with the simulated annealing blind deconvolution algorithm. Either of these schemes for varying α may well prove to be more effective than that employed by Nieto-Vesperinas *et al.* (1988).

Varying α automatically via (6.9) appears to be preferable to setting $\alpha_{n_c} = c_3\sqrt{T_{n_c}}$, since the need to select c_3 is thereby eliminated. The example presented in Figure 6.5 indicates that the algorithm operates satisfactorily when adjusting α such that the value of p_{acc} is maintained in the range $[0.4, 0.6]$. However, it would be worthwhile to perform computational experiments to determine whether this is the optimal range for p_{acc} to lie within.

The examples presented in Figure 6.6, in which quenching is employed, confirm that $Q\{\hat{f},\hat{h}\}$ does indeed possess local minima. It is expected that this should be so, for the reason stated in §6.2. Note that when quenching with the second pair of starting images, the local minimum that is found has a sufficiently small value of $Q\{\hat{f},\hat{h}\}$ that the corresponding image estimates are faithful reconstructions of the original images, as is evident from Figure 6.7b. However, computational experience suggests that when operating on convolutions which comprise a significantly greater number of pixels than Figure 6.2c, quenching is usually incapable of finding solutions with usefully small values of $Q\{\hat{f},\hat{h}\}$. This accords with the findings of Nieto-Vesperinas (1986), who shows that the number of local minima possessed by the cost function associated with the Fourier phase problem increases dramatically with the image size.

Figure 6.8 demonstrates that the simulated annealing blind deconvolution algorithm performs in a predictable manner when applied to contaminated convolution data. The cost function $Q\{\hat{f},\hat{h}\}$ is steadily reduced, until it reaches a value slightly less than the contamination level. This is, of course, the best that one can hope to achieve. Notice the noise that is apparent in the reconstructed images presented in Figure 6.9, which is a consequence of the contamination on the convolution data. It may well be possible to reduce the level of such noise by employing the strategy described by Navarro *et al.* (1989), in which T is gradually reduced to some prescribed value (which depends upon the estimated contamination level), after which the image estimates obtained at the end of each of a number of image scans are averaged.

The simulated annealing blind deconvolution algorithm is undeniably computationally intensive, the example presented in Figure 6.10a,b requiring approximately 25 minutes of CPU time on a VAX 3500. It is thus worthwhile attempting to reduce the computational requirements of the algorithm. It would be advantageous to only perform enough scans in each cycle to ensure that the "system" is in "equilibrium". Both Nieto-Vesperinas *et al.* (1988) and Sechen (1988, §2.4) describe methods for determining whether equilibrium has been reached. Furthermore, it may be possible (see e.g. Sechen, 1988, §§2.3,2.4, and references quoted therein) to devise annealing schedules which are optimal, or nearly so.

Recall from §6.1 that it would be desirable to incorporate a regularization scheme into the simulated annealing blind deconvolution algorithm, in an attempt to ameliorate the effects of the ill-posed nature of the blind deconvolution problem. The averaging scheme mentioned in the paragraph before last can be looked upon as a crude regularization scheme. A more elegant method of incorporating a regularization scheme into the simulated annealing blind deconvolution algorithm is to include a regularization term in the cost function. Instead of minimizing $Q\{\hat{f},\hat{h}\}$, one would employ Algorithm 6.1 to minimize the cost function $Q_2\{\hat{f},\hat{h}\}$, where $Q_2\{\hat{f},\hat{h}\}$ is defined by

$$Q_2\{\hat{f},\hat{h}\} = (1-\mu)\,Q\{\hat{f},\hat{h}\} + \mu\,(R\{\hat{f}\} + R\{\hat{h}\}) \tag{6.10}$$

and $R\{\hat{f}\}$ is some regularization function. The constant μ, which satisfies $0 \leq \mu \leq 1$, determines the relative strength of the regularization constraint.

One possible form of $R\{\hat{f}\}$ is

$$R\{\hat{f}\} = \sum_{m,n:M_f[m,n]=1}\sum \left(\left|\hat{f}[m+1,n] - \hat{f}[m,n]\right|^{\gamma} + \left|\hat{f}[m,n+1] - \hat{f}[m,n]\right|^{\gamma} \right) \tag{6.11}$$

where γ is a positive constant (typically either 1 or 2), with $R\{\hat{h}\}$ being similarly defined. Note particularly that, when implementing Algorithm 6.1 with $Q\{\hat{f},\hat{h}\}$ replaced by $Q_2\{\hat{f},\hat{h}\}$, the evaluation of the change in $Q_2\{\hat{f},\hat{h}\}$, denoted by ΔQ_2, in step 4.3 is computationally straightforward if $R\{\hat{f}\}$ is that defined by (6.11). This is because only three terms of the summation in (6.11) are modified when one pixel of $\hat{f}[m,n]$ is perturbed. The evaluation of ΔQ_2 is thus only minimally more computationally demanding than the evaluation of ΔQ.

The simulated annealing blind deconvolution algorithm, in which a regularization scheme is incorporated along the lines described in the previous paragraph, promises to be a robust techniques for effecting the blind deconvolution of contaminated convolution data. However, this algorithm is computationally intensive, and so

it may be worthwhile to investigate the use of an appropriate combination of the iterative blind deconvolution algorithm and the simulated annealing blind deconvolution algorithm. The speed of the iterative blind deconvolution algorithm could thereby be combined with the robustness of the simulated annealing blind deconvolution algorithm, producing what may well prove to be an effective blind deconvolution strategy.

Chapter 7

Blind Deconvolution of Impulsive Images

There are situations, as explained in §7.1, wherein one is presented with the convolution of two images, each comprising a collection of isolated points. Such images are here termed *impulsive images*. This chapter describes a new blind deconvolution technique, which is applicable when both components of a given convolution are impulsive.

The motivation behind the development of algorithms specifically for the blind deconvolution of impulsive images is given in §7.1. A simple blind deconvolution algorithm is described in §7.2, which is applicable to convolutions with impulsive components that bear no special relationship to one another. In §7.3, an algorithm is described which effects the blind deconvolution of two impulsive images, applicable in situations where it is known *a priori* that the locations of the points of one component are the same as those of the other component, apart from being rotated by 180°. It should be pointed out that ghosted shift-and-add images of star clusters (see §2.8.2.2) are precisely of this form. Further discussion of the limitations of, and possible improvements to, the new blind deconvolution algorithms described in this chapter are discussed in §7.4.

7.1 Motivation

As pointed out in §2.8.2.2, shift-and-add images of star clusters in optical astronomy often exhibit pronounced ghosting, which can be adequately modelled (after the fog has been removed) as the convolution of the true image with a ghosting psf. This is an example of a situation in which one wishes to blindly deconvolve the convolution of two images, both of which are impulsive. The iterative blind deconvolution algorithm described in §3.3.3.2, or the simulated annealing algorithm of §6.2, can of course be applied to blindly deconvolve impulsive images.

However, the "golden rule" of image reconstruction is that one should always make use of all the *a priori* information available regarding the image (or images) to be reconstructed (e.g. Bates and McDonnell, 1986, §52; Trussell and Civanlar, 1984). As shown in §2.7, the only constraint necessary to ensure the existence of a unique solution to any multi-dimensional blind deconvolution problem (provided the data are

uncontaminated) is that both convolution components must possess finite extents. But further constraints on the images are invaluable in practice, both for reducing the number of possible solutions (i.e. those which are consistent, in a mean-square sense, with the data) when the convolution data are contaminated, and for speeding the convergence of iterative solution algorithms. When presented with a blind deconvolution problem in which it is known that both components are impulsive, it makes sense to incorporate this knowledge into the reconstruction scheme being employed. Note that it is straightforward, in practice, to determine if the components of a given convolution are impulsive, because in such cases the convolution must also be impulsive.

To incorporate impulsiveness into the iterative blind deconvolution algorithm described in §3.3.3.2, one must first express this *a priori* knowledge as a constraint, and then determine a means of enforcing this constraint. Since the theory of projections onto convex sets (see §3.1.1.2) is later invoked, it is convenient to define the impulsiveness constraint in terms of a set of images which satisfy the constraint. The set of impulsive images comprising N non-zero points, denoted by $\mathcal{S}^N_{\text{impulsive}}$, is here defined as

$$\mathcal{S}^N_{\text{impulsive}} = \{f(\vec{x}) \; : \; f(\vec{x}) = \sum_{n=1}^{N} a_n \, \delta(\vec{x} - \vec{x}_n)\} \tag{7.1}$$

where $a_1, \ldots, a_N$ and $\vec{x}_1, \ldots, \vec{x}_N$ are arbitrary. Note that this is but one possible definition of impulsive images. It is evident that a linear combination of two impulsive images, namely $f_1(\vec{x})$ and $f_2(\vec{x})$, which is denoted by $f_3(\vec{x})$ and is given by

$$f_3(\vec{x}) = \alpha f_1(\vec{x}) + (1-\alpha) f_2(\vec{x}), \quad 0 < \alpha < 1, \quad f_1(\vec{x}), f_2(\vec{x}) \in \mathcal{S}^N_{\text{impulsive}} \tag{7.2}$$

is not necessarily a member of $\mathcal{S}^N_{\text{impulsive}}$, since $f_3(\vec{x})$ generally comprises $2N$ non-zero points. The set $\mathcal{S}^N_{\text{impulsive}}$ is therefore not a convex set, and so the convergence of any algorithm employing projections onto this set is not guaranteed. Since it is, at present, unclear whether the iterative blind deconvolution algorithm of §3.3.3.2 can be described in terms of projections onto convex sets, one can merely conjecture that the convergence of this algorithm may be impeded by the incorporation of an impulsiveness constraint.

To indicate how the impulsiveness constraint may be applied to the iterative blind deconvolution algorithm, it is necessary to consider the projection (see §3.1.1.2) onto the set $\mathcal{S}^N_{\text{impulsive}}$. It then follows that the impulsiveness constraint may be applied, at each iteration of the iterative blind deconvolution algorithm, to the estimate $\hat{f}[m,n]$ of $f[m,n]$, thereby forming $\hat{f}_c[m,n]$, by setting

$$\hat{f}_c[m',n'] = \begin{cases} \hat{f}[m',n'] & \text{for } [m',n'] : \hat{f}[m',n'] \text{ is one of the} \\ & \qquad N \text{ brightest pixels of } f[m,n] \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

The discrete image notation is here invoked to show how the constraint is computed in practice. The estimate $\hat{h}[m,n]$ of $h[m,n]$ may be constrained in a similar manner.

It is now instructive to speculate why such a constraint may be detrimental to the convergence of the iterative blind deconvolution algorithm (or indeed any iterative algorithm employing such a constraint). Consider a particular pixel of $\hat{f}[m,n]$, located

at $[m', n']$, which corresponds to a non-zero pixel in $f[m, n]$. If it so happens, at a particular iteration, that $\hat{f}[m', n']$ is not one of the N brightest pixels of $\hat{f}[m, n]$ (as could easily occur if the iterations begin with a pseudo-random $\hat{f}[m, n]$), then $\hat{f}_c[m', n']$ is set to zero. If such a situation persists for a number of iterations, $\hat{f}[m, n]$ may be prohibited from reaching its desired value, thereby preventing convergence of the algorithm.

One way in which an impulsiveness constraint could be usefully included in the iterative blind deconvolution algorithm is by applying this constraint only after a pre-chosen number of iterations have been performed. Alternatively, a hybrid input-output scheme (see §3.2.8.2) could be devised, whereby the reconstructed images are "urged" to satisfy the impulsiveness constraint, rather than being forced to do so. Note that in either case, the value of N must somehow be estimated. A further possibility is to choose a value for N which is significantly greater than the expected number of non-zero pixels in the original image, in the hope of avoiding the situation described in the previous paragraph. Alternatively, the value of N could be gradually decreased as the iterations proceed.

Thus, when given a blind deconvolution problem, in which it is known that the convolution components are impulsive, it can be difficult to incorporate this valuable *a priori* knowledge into the iterative blind deconvolution algorithm described in §3.3.3.2. So it is worthwhile seeking alternative methods to solve such blind deconvolution problems. Recall now the autocorrelation tri-intersection method, discussed in §3.2.3.2, which directly (i.e. non-iteratively) effects the reconstruction of impulsive images from their autocorrelations. It transpires that this method, which implicitly includes an impulsiveness constraint, can be appropriately adapted to blindly deconvolve impulsive images, as is described and demonstrated in §7.2 and §7.3.

7.2 Blind Deconvolution of General Impulsive Images

This section gives the theory behind the proposed methods for blindly deconvolving the convolution of two impulsive images. It is assumed here that no special relationship exists between the positions of the points of the two components. From now on, *point* is taken to mean any part of the image in question which has non-zero value, or, in the case of contaminated images, any part which has value greater than the background level. The blind deconvolution of ghosted shift-and-add images, a situation in which the convolution components are closely related, is dealt with separately in §7.3.

Consider an image $f(\vec{x})$ comprising M_f isolated points, the m^{th} of which is located at $\vec{\alpha}_m$ with amplitude f_m. It is convenient to represent $f(\vec{x})$ by

$$f(\vec{x}) = \sum_{m=1}^{M_f} f_m \, \delta(\vec{x} - \vec{\alpha}_m) \tag{7.4}$$

Another impulsive image, $h(\vec{x})$, is represented by

$$h(\vec{x}) = \sum_{m=1}^{M_h} h_m \, \delta(\vec{x} - \vec{\beta}_m) \tag{7.5}$$

The convolution of $f(\vec{x})$ and $h(\vec{x})$, denoted by $g(\vec{x})$, is then given by

$$\begin{aligned} g(\vec{x}) &= f(\vec{x}) \odot h(\vec{x}) \\ &= \sum_{m=1}^{M_f} \sum_{n=1}^{M_h} f_m \, h_n \, \delta(\vec{x} - \vec{\alpha}_m - \vec{\beta}_n) \end{aligned} \tag{7.6}$$

Thus $g(\vec{x})$ possesses (at most) $M_f \, M_h$ distinct points, and comprises M_h translates (as defined in §3.2.3.2) of $f(\vec{x})$, or alternatively, comprises M_f translates of $h(\vec{x})$.

The similarity between (7.6) and (3.25) (the expression for the autocorrelation $a(\vec{x})$ of $f(\vec{x})$) suggests that it may be possible to recover $f(\vec{x})$ or $h(\vec{x})$ from $g(\vec{x})$, in a similar manner to that described in §3.2.3.2 for recovering $f(\vec{x})$ from $a(\vec{x})$. The autocorrelation tri-intersection method of §3.2.3.2 involves combining translates of $a(\vec{x})$, in such a manner that translates of $f(\vec{x})$ (or $f(-\vec{x})$) contained in $a(\vec{x})$ are superimposed. The general idea behind the blind deconvolution techniques described in this chapter is to appropriately combine translates of $g(\vec{x})$, so that translates of $f(\vec{x})$ (or $h(\vec{x})$) contained in $g(\vec{x})$ are superimposed. To this end, the convolution product $g^{(2)}(\vec{x})$ is formed, as

$$g^{(2)}(\vec{x}) = g(\vec{x}) \, g(\vec{x} - \vec{s}) \tag{7.7}$$

where $\vec{s}$, which is here termed a *shift vector*, must be chosen to satisfy either

$$\vec{s} = \vec{\alpha}_k - \vec{\alpha}_l, \quad k \neq l, \quad \vec{\alpha}_k, \vec{\alpha}_l \in S_f \tag{7.8}$$

or

$$\vec{s} = \vec{\beta}_k - \vec{\beta}_l, \quad k \neq l, \quad \vec{\beta}_k, \vec{\beta}_l \in S_h \tag{7.9}$$

where S_f and S_h denote the supports (see §3.2.3.2) of $f(\vec{x})$ and $h(\vec{x})$ respectively. If $\vec{s}$ is of the form (7.9), then substitution of (7.9) and (7.6) into (7.7) yields

$$g^{(2)}(\vec{x}) = \sum_{m=1}^{M_f} \sum_{n=1}^{M_h} f_m \, h_n \, \delta(\vec{x} - \vec{\alpha}_m - \vec{\beta}_n) \sum_{p=1}^{M_f} \sum_{q=1}^{M_h} f_p \, h_q \, \delta(\vec{x} - \vec{\alpha}_p - \vec{\beta}_q - \vec{\beta}_k + \vec{\beta}_l) \tag{7.10}$$

Note that $g^{(2)}(\vec{x})$ only contains points at positions for which values of m, n, p, q, k and l exist that satisfy

$$\vec{x} = \vec{\alpha}_m + \vec{\beta}_n = \vec{\alpha}_p + \vec{\beta}_q + \vec{\beta}_k - \vec{\beta}_l \tag{7.11}$$

It is evident that (7.11) is satisfied when

$$m = p \quad \text{and} \quad n = k \quad \text{and} \quad q = l \tag{7.12}$$

Thus (7.10) can be expressed as

$$g^{(2)}(\vec{x}) = h_k \, h_l \sum_{m=1}^{M_f} f_m^2 \, \delta(\vec{x} - \vec{\alpha}_m - \vec{\beta}_k) + \text{other terms} \tag{7.13}$$

from which it follows that $g^{(2)}(\vec{x})$ contains a translate of $[f(\vec{x})]^2$. The other terms in (7.13) arise if any further combinations of m, n, p, q, k, and l exist which satisfy (7.11), apart from those in (7.12). The conditions under which no other terms appear in (7.13) can be precisely stated as

$$\begin{array}{l} \text{whenever} \quad \vec{\alpha}_a, \vec{\alpha}_b \in S_f \quad \text{and} \quad \vec{\beta}_c, \vec{\beta}_d, \vec{\beta}_e, \vec{\beta}_f \in S_h \\ \text{and} \quad \vec{\alpha}_a - \vec{\alpha}_b + \vec{\beta}_c - \vec{\beta}_d + \vec{\beta}_e - \vec{\beta}_f = \vec{0} \\ \text{then} \quad a = b \quad \text{and} \quad c = d \quad \text{and} \quad e = f \\ \text{or} \quad a = b \quad \text{and} \quad c = f \quad \text{and} \quad d = e \end{array} \tag{7.14}$$

The nonredundancy condition (7.14) is violated if the points contained in any subsets of S_f and S_h happen to be arranged in any of the following three manners:

1. A pair of points in $f(\vec{x})$ has the same vector separation as a pair of points in $h(\vec{x})$.
2. Two pairs of points in $h(\vec{x})$ possess the same vector separation.
3. The vector separation of a pair of points in $f(\vec{x})$ is equal to the sum of the vector separations of two pairs of points in $h(\vec{x})$.

If $\vec{s}$ is of the form (7.8), rather than (7.9), then $g^{(2)}(\vec{x})$ is given by

$$g^{(2)}(\vec{x}) = f_k \, f_l \sum_{m=1}^{M_h} h_m^2 \, \delta(\vec{x} - \vec{\beta}_m - \vec{\alpha}_k) + \text{other terms} \tag{7.15}$$

rather than by (7.13). The other terms in (7.15) appear if a nonredundancy condition, which is identical to (7.14) apart from S_f and S_h being interchanged, is violated.

When reconstructing $f(\vec{x})$ from its autocorrelation, it is straightforward to choose a suitable $\vec{s}$ by direct inspection of $a(\vec{x})$, as explained in §3.2.3.2. Unfortunately, it is not possible to choose $\vec{s}$ directly from inspection of $g(\vec{x})$. However, if $\vec{s}$ is selected appropriately, so that a translate of $f(\vec{x})$ (or of $h(\vec{x})$) in $g(\vec{x})$ overlaps with another translate in $g(\vec{x} - \vec{s})$, then the correlation between $g(\vec{x})$ and $g(\vec{x} - \vec{s})$ must have significant value. This suggests that the autocorrelation of $g(\vec{x})$, denoted by $gg(\vec{x})$, may assist in the selection of a suitable $\vec{s}$. To make clear why this is so, $gg(\vec{x})$ is expressed as

$$\begin{aligned} gg(\vec{x}) &= g(\vec{x}) \odot g(-\vec{x}) \\ &= [f(\vec{x}) \odot h(\vec{x})] \odot [f(-\vec{x}) \odot h(-\vec{x})] \\ &= [f(\vec{x}) \odot f(-\vec{x})] \odot [h(\vec{x}) \odot h(-\vec{x})] \\ &= ff(\vec{x}) \odot hh(\vec{x}) \end{aligned} \tag{7.16}$$

The autocorrelation $gg(\vec{x})$ of the convolution $g(\vec{x})$ is thus equal to the convolution of the autocorrelations of $f(\vec{x})$ and $h(\vec{x})$. It is useful to separate $ff(\vec{x})$ into two parts, namely that at the origin and the remainder, as

$$ff(\vec{x}) = ff_0 \, \delta(\vec{x}) + ff'(\vec{x}) \tag{7.17}$$

where, from (3.25), ff_0 is given by

$$ff_0 = \sum_{m=1}^{M_f} f_m^2 \tag{7.18}$$

In the same manner, $h(\vec{x})$ is separated, with hh_0 and $hh'(\vec{x})$ being similarly defined. By substituting (7.17) into (7.16), $gg(\vec{x})$ can be expressed as

$$gg(\vec{x}) = ff_0\, hh_0\, \delta(\vec{x}) + hh_0\, ff'(\vec{x}) + ff_0\, hh'(\vec{x}) + ff'(\vec{x}) \odot hh'(\vec{x}) \tag{7.19}$$

Thus $gg(\vec{x})$ contains $ff'(\vec{x})$ (scaled by hh_0) and $hh'(\vec{x})$ (scaled by ff_0), along with other terms. Figure 7.1 illustrates this interesting property of the autocorrelation of an impulsive convolution. Note for each of $gg(\vec{x})$, $ff(\vec{x})$, and $hh(\vec{x})$, the pixel at the origin has significantly greater value than all other pixels. So as to make all pixels visually apparent, the pixel at the origin, in each of the images shown in Figure 7.1d–f, has been set equal to the value of the pixel whose magnitude is closest to that of the pixel at the origin.

Furthermore, it follows from (3.25) and (7.18) that

$$hh_0 > hh'(\vec{x}) \quad \text{and} \quad ff_0 > ff'(\vec{x}) \quad \forall \vec{x} \tag{7.20}$$

The translates of $ff'(\vec{x})$ and $hh'(\vec{x})$ in $gg(\vec{x})$, contained in the second and third terms of (7.19) respectively, must therefore be larger than any of those resulting from the last term of (7.19).

Recall now, from §3.2.3.2, that the vector from the origin to any other point in $ff(\vec{x})$ corresponds to the vector separation of two points in $f(\vec{x})$ (the positions of points in $hh(\vec{x})$ being similarly related to the vector separations of points in $h(\vec{x})$). Thus the vector from the origin to the point in $gg(\vec{x})$ with the largest amplitude (apart from the origin) must be equal to the vector separation of two points in either $f(\vec{x})$ or $h(\vec{x})$, and so it is a suitable choice for $\vec{s}$ in (7.7), since it evidently satisfies either (7.8) or (7.9).

An algorithm to effect the blind deconvolution of impulsive images, based on this technique, is described in the next subsection.

7.2.1 Description of Algorithm

A simple algorithm to blindly deconvolve the convolution of two impulsive images, based on the technique described in §7.2, is now presented. Although this algorithm is somewhat unrefined, it nonetheless demonstrates the potential of this technique. Some of the considerations made when devising the algorithm are first discussed, before listing the steps in the algorithm.

As shown in §7.2, the convolution product $g^{(2)}(\vec{x})$ is guaranteed to contain no spurious points (i.e. points other than those contained in a translate of either $f(\vec{x})$ or $h(\vec{x})$) if condition (7.14) is satisfied. Note, however, that $g^{(2)}(\vec{x})$ does not necessarily contain spurious points if condition (3.33) is violated; whether or not it does depends upon the particular choice of $\vec{s}$. Spurious points appearing in $g^{(2)}(\vec{x})$ can often be removed by appropriately combining a number of different $g^{(2)}(\vec{x})$, each of which satisfies (7.7) for a different value of $\vec{s}$. Some of the difficulties encountered when doing so are now discussed.

It is evident, from (7.16), that a point in $gg(\vec{x})$ at $\vec{s}$ is accompanied by another point of equal amplitude at $-\vec{s}$, since $gg(\vec{x})$ is centro-symmetric. Replacing $\vec{s}$ by $-\vec{s}$ in (7.7) merely results in k and l being interchanged in (7.13), so it is clear that,

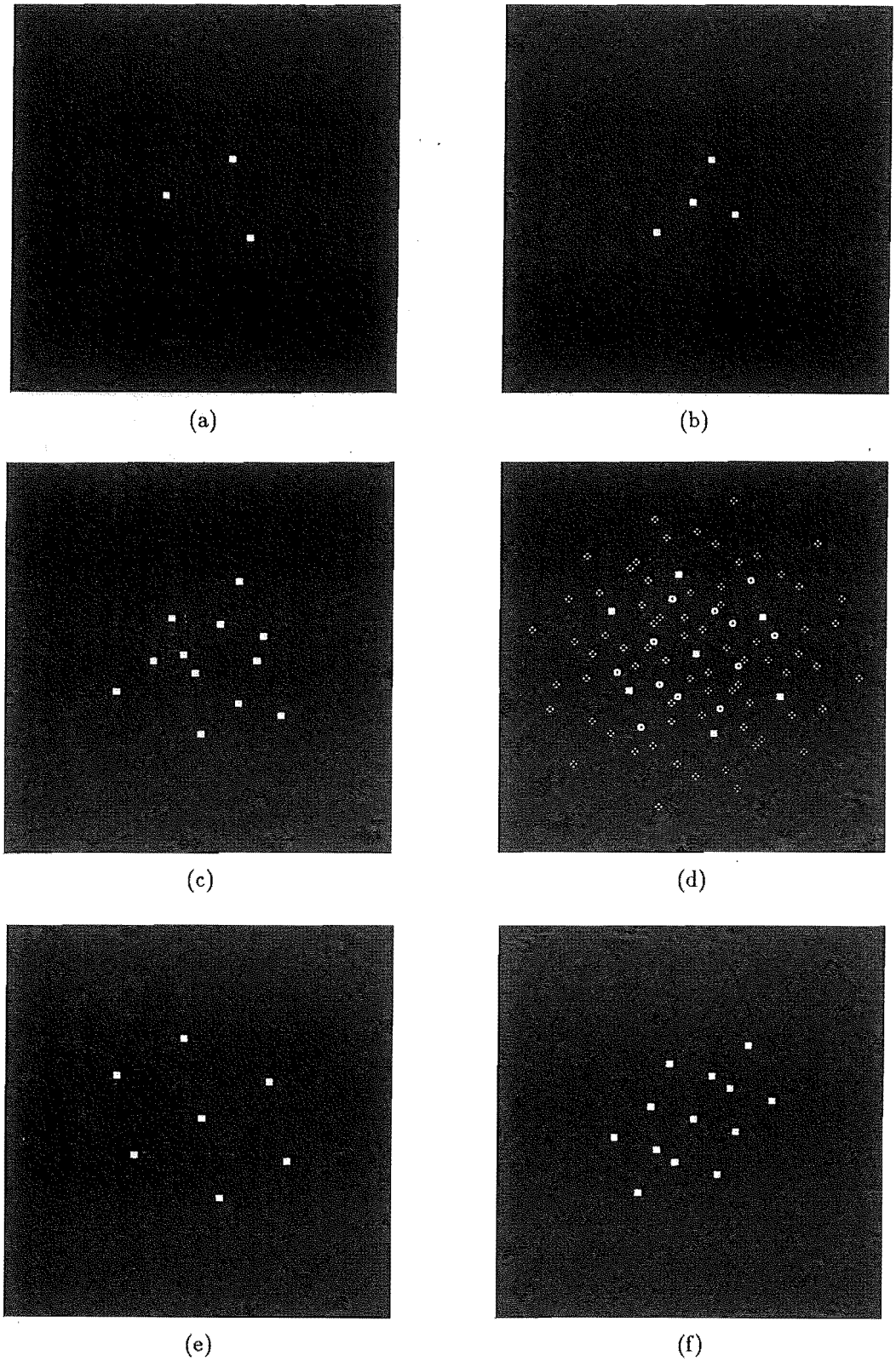


Figure 7.1 Illustration of the autocorrelation of a convolution. (a,b) original images $f(\vec{x})$ and $h(\vec{x})$, (c) convolution $g(\vec{x}) = f(\vec{x}) \odot h(\vec{x})$, (d) autocorrelation $gg(\vec{x})$ of $g(\vec{x})$, (e) autocorrelation $ff(\vec{x})$ of $f(\vec{x})$, and (f) autocorrelation $hh(\vec{x})$ of $h(\vec{x})$. Note that the pixel at the origin in each of (d), (e), and (f) has been modified as described in the text following (7.19).

when computing various $g^{(2)}(\vec{x})$, only one member of each such pair of points in $gg(\vec{x})$ should be selected for $\vec{s}$. It is convenient to denote by $\vec{s}_j$ one member of the pair of points in $gg(\vec{x})$ having the j^{th} largest amplitude. The convolution product computed by (7.7) with $\vec{s} = \vec{s}_j$ is then denoted by $g_j^{(2)}(\vec{x})$.

Unfortunately, there is no guarantee that say $g_1^{(2)}(\vec{x})$ and $g_2^{(2)}(\vec{x})$ both contain a translate of the same component of $g(\vec{x})$ (i.e. both contain a translate of $f(\vec{x})$, or both contain a translate of $h(\vec{x})$). However, if one component happens to contain significantly more points than the other component, then it is likely that at least the first few $g_j^{(2)}(\vec{x})$ (i.e. any $g_j^{(2)}(\vec{x})$ for which j is small) contain translates of that component, due to the relative values of ff_0 and hh_0 in (7.19). In the algorithm that follows, it is assumed that this is so (but mention is made, in §7.2.2, of how this assumption can be avoided).

The recovered convolution component in each $g_j^{(2)}(\vec{x})$ is likely to be situated in different positions in image-space, as is evident from (7.13). Thus each $g_j^{(2)}(\vec{x})$ must be appropriately aligned before being combined. If for various j, each $g_j^{(2)}(\vec{x})$ contains spurious points, then it is unlikely that the spurious points in all $g_j^{(2)}(\vec{x})$ are similarly juxtaposed with the points of the recovered component. This is the reasoning behind combining a number of $g_j^{(2)}(\vec{x})$. The steps of the algorithm are now listed:

Algorithm 7.1

Step 1: Form $gg(\vec{x}) = g(\vec{x}) \odot g(-\vec{x})$.

Step 2: Set $gg(\vec{0}) = 0$.

Step 3: Form binary mask $g_{\text{b}}(\vec{x}) = \begin{cases} 1, & \vec{x} : g(\vec{x}) > 0 \\ 0, & \vec{x} : g(\vec{x}) = 0 \end{cases}$.

Step 4: Repeat the following steps, for $j = 1$ to J:

- Step 4.1: Find $\vec{s}_j$, the value of $\vec{x}$ at which $gg(\vec{x})$ is maximum.
- Step 4.2: Set $gg(\vec{s}_j) = 0$, and set $gg(-\vec{s}_j) = 0$.
- Step 4.3: If $j = 1$, form $\tilde{f}_1(\vec{x}) = g(\vec{x})\, g_b(\vec{x} - \vec{s}_1)$.
- Step 4.4: If $j > 1$, perform the following three steps:
  - Step 4.4.1: Form $p_j(\vec{x}) = g_{\text{b}}(\vec{x})\, g_b(\vec{x} - \vec{s}_j)$.
  - Step 4.4.2: Find the $\vec{\gamma}$ which maximizes the correlation between $p_j(\vec{x} - \vec{\gamma})$ and $\tilde{f}_{j-1}(\vec{x})$.
  - Step 4.4.3: Form $\tilde{f}_j(\vec{x}) = p_j(\vec{x} - \vec{\gamma})\, \tilde{f}_{j-1}(\vec{x})$.

Step 5: Set $\tilde{f}(\vec{x}) = \tilde{f}_J(\vec{x})$.

Some aspects of the above algorithm require further explanation. The *intermediate image estimate* formed by combining j convolution products is denoted by $\tilde{f}_j(\vec{x})$, the final estimate being $\tilde{f}(\vec{x})$ (but note that $\tilde{f}(\vec{x})$ may, in fact, be an estimate of $h(\vec{x})$, rather than of $f(\vec{x})$). The binary mask $g_{\text{b}}(\vec{x})$ is formed so that $\tilde{f}_j(\vec{x})$ contains an estimate of $f(\vec{x})$ or $h(\vec{x})$, rather than an estimate of $f(\vec{x})$ or $h(\vec{x})$ raised to the power of $2J$. Thus $\tilde{f}_1(\vec{x})$ is equivalent to $[g_1^{(2)}(\vec{x})]^{1/2}$, and the *binary convolution product*

$p_j(\vec{x})$ is equivalent to a binarized version of $g_j^{(2)}(\vec{x})$. Step 4.4.2 aligns the j^{th} binary convolution product with the $(j-1)^{\text{th}}$ intermediate image estimate.

It should be emphasized that $\tilde{f}(\vec{x})$ is only guaranteed to contain a faithful estimate of $f(\vec{x})$ (or $h(\vec{x})$) if condition (7.14) is satisfied (in which case $J = 1$ is sufficient, as shown in §7.2). If condition (7.14) is violated, then an appropriate choice of J produces, in most cases, a $\tilde{f}(\vec{x})$ which contains the correct number of points. However, even if $\tilde{f}(\vec{x})$ contains the correct number of points, the amplitudes of some points may be grossly in error. This occurs if points in $g(\vec{x})$ happen to be overlapped (i.e. if $g(\vec{x})$ contains less then $M_f M_h$ distinct points).

Once $\tilde{f}(\vec{x})$ has been computed by Algorithm 7.1, it is then straightforward to compute an estimate of the other convolution component, denoted by $\tilde{h}(\vec{x})$. This is achieved by forming convolution products as before, but with each $\vec{s}_j$ chosen to be the vector separation between various pairs of points in $\tilde{f}(\vec{x})$. The steps of this algorithm are now listed:

Algorithm 7.2

Step 1: Form binary mask $g_b(\vec{x}) = \begin{cases} 1, & \vec{x} : g(\vec{x}) > 0 \\ 0, & \vec{x} : g(\vec{x}) = 0 \end{cases}$.

Step 2: Repeat the following steps, for $j = 0$ to J:

Step 2.1: Find $\vec{x}_j$, the value of $\vec{x}$ at which $\tilde{f}(\vec{x})$ is maximum.

Step 2.2: Set $\tilde{f}(\vec{x}_j) = 0$.

Step 2.3: If $j > 0$, set $\vec{s}_j = \vec{x}_0 - \vec{x}_j$.

Step 2.4: If $j = 1$, form $\tilde{h}_1(\vec{x}) = g(\vec{x})\, g_b(\vec{x} - \vec{s}_j)$.

Step 2.5: If $j > 1$, form $\tilde{h}_j(\vec{x}) = \tilde{h}_{j-1}(\vec{x})\, g_b(\vec{x} - \vec{s}_j)$.

Step 3: Set $\tilde{h}(\vec{x}) = \tilde{h}_J(\vec{x})$.

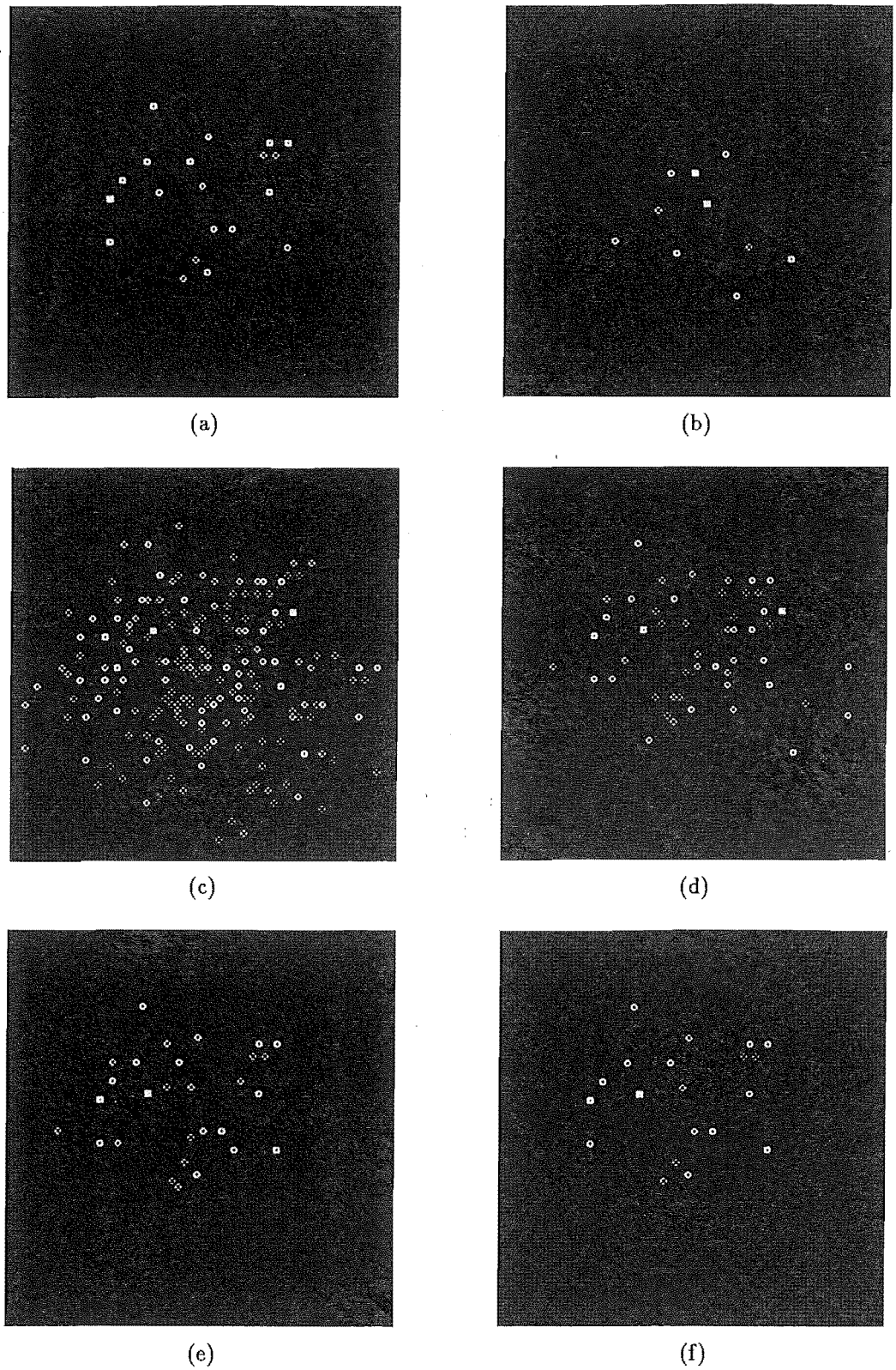
Algorithm 7.2 is noticeably simpler than Algorithm 7.1, since it makes use of the previously computed $\tilde{f}(\vec{x})$. Figure 7.2 illustrates the application of Algorithms 7.1 and 7.2 to a convolution comprising a large number of points ($M_f = 20$, $M_h = 10$). Although condition (7.14) is violated by a number of points in this example, the locations of all the points of both components are determined correctly, and no spurious points remain in either final estimate. Note, however, that the amplitudes of some of the points are in error.

The shortcomings of, and possible improvements to, these algorithms are discussed in the next subsection.

7.2.2 Improvements to Algorithm

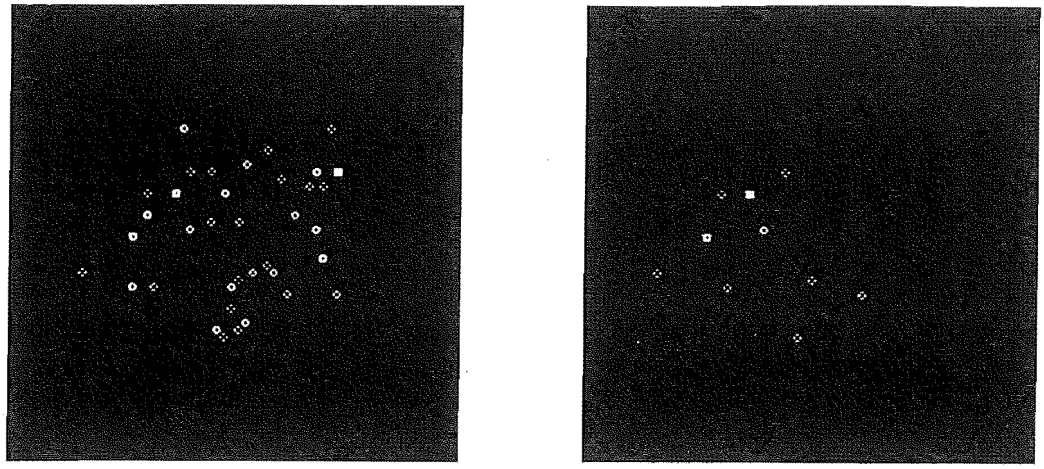
Some of the shortcomings of Algorithms 7.1 and 7.2 have already been mentioned in §7.2.1. Possible ways of overcoming these and other problems are now discussed.

The uncertainty as to which convolution component is contained in $p_j(\vec{x})$ (or, equivalently, in $g_j^{(2)}(\vec{x})$) in Algorithm 7.1 has been mentioned previously. This problem



(a) (b) (c) (d) (e) (f)

Figure 7.2 Figure continued on following page.



(g) (h)

Figure 7.2 Illustration of blind deconvolution of general impulsive images (Algorithms 7.1 and 7.2). (a,b) original images $f(\vec{x})$ and $h(\vec{x})$, (c) convolution $g(\vec{x}) = f(\vec{x}) \odot h(\vec{x})$, recovered component $\tilde{f}(\vec{x})$ for (d) $J = 1$, (e) $J = 2$, and (f) $J = 3$, recovered component $\tilde{h}(\vec{x})$ for (g) $J = 1$ and (h) $J = 2$.

can be overcome by including an extra step in the algorithm, in which the number of points that $p_j(\vec{x} - \vec{\gamma})$ and $\tilde{f}_{j-1}(\vec{x})$ have in common is computed. If this number is less than some pre-determined threshold, then $p_j(\vec{x})$ and $\tilde{f}_{j-1}(\vec{x})$ are deemed to contain different convolution components, and so $p_j(\vec{x})$ is discarded (i.e. set $\tilde{f}_j(\vec{x}) = \tilde{f}_{j-1}(\vec{x})$).

The value of J (i.e. the number of convolution products used to produce $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$) in Algorithms 7.1 and 7.2 must somehow be chosen. One possibility is simply to continue combining convolution products until the number of points in the image estimate remains unchanged; i.e. until $\tilde{f}_j(\vec{x}) = \tilde{f}_{j-1}(\vec{x})$ in Algorithm 7.1, or $\tilde{h}_j(\vec{x}) = \tilde{h}_{j-1}(\vec{x})$ in Algorithm 7.2.

As mentioned previously, the amplitudes of some points in $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$ may be in error. Whether or not this is so can be determined by computing the difference between $\tilde{f}(\vec{x}) \odot \tilde{h}(\vec{x})$ and $g(\vec{x})$ (but note that $\tilde{f}(\vec{x}) \odot \tilde{h}(\vec{x})$ must first be scaled so as to have the same energy as $g(\vec{x})$). Methods for correcting the amplitudes of points in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ are discussed further in §7.3.2.

In practice, any impulsive convolution is necessarily contaminated, and contains points of finite diameter. The difficulties resulting from such practicalities are dealt with in §§7.3 and 7.4. This discussion of Algorithms 7.1 and 7.2 is somewhat brief, because the main application presently envisaged for such algorithms is the blind deconvolution of ghosted shift-and-add images. Moreover, it transpires that a significantly simpler algorithm than Algorithm 7.2 results from consideration of the relationship between the components of a ghosted shift-and-add image, as detailed in the following section.

7.3 Blind Deconvolution of Shift-And-Add Images

The mechanism by which ghosts are formed by the shift-and-add algorithm has been described in §2.8.2.2, where it was shown how the shift-and-add image can be described as a convolution. It is convenient now to denote by $g(\vec{x})$ the defogged shift-and-add image ($saa_{\rm d}(x,y)$ in §2.8.2.1) resulting from the original image $f(\vec{x})$, and to denote the object dependent part of the shift-and-add psf ($h^{\circ}_{\rm saa}(x,y)$ in §2.8.2.2) by $h(\vec{x})$. From (2.46), it follows that these quantities are related by

$$g(\vec{x}) = f(\vec{x}) \odot h(\vec{x}) + c(\vec{x}) \tag{7.21}$$

where $c(\vec{x})$ incorporates all departures of $g(\vec{x})$ from the simple convolutional model. As in §7.2, the true image is represented by

$$f(\vec{x}) = \sum_{m=1}^{M_f} f_m \, \delta(\vec{x} - \vec{\alpha}_m) \tag{7.22}$$

(which is identical to (7.4)). In practice, a shift-and-add image $saa(x,y)$ (as defined in §2.8.2.1) comprises points of finite diameter (roughly the size of the Airy disk). However, the defogged shift-and-add image $saa_{\rm d}(x,y)$ tends to have points of significantly smaller diameter than those in $saa(x,y)$. This is because the central peak in the shift-and-add reference image $saa^{\rm r}(x,y)$ is about the same size as the points in $saa(x,y)$, and so the points in $saa_{\rm d}(x,y)$ tend to be sharpened by the defogging operation. It is thus reasonable to also model $h(\vec{x})$ as a sum of delta functions. From the discussion in §2.8.2.2 of the nature of $h^{\circ}_{\rm saa}(x,y)$, it is evident that $h(\vec{x})$ must be of the form

$$h(\vec{x}) = \sum_{m=1}^{M_f} h_m \, \delta(\vec{x} + \vec{\alpha}_m) \tag{7.23}$$

as compared to the general form in (7.5).

It is worthwhile mentioning the relationship between the values of h_m and f_m in (7.23) and (7.22) respectively. To do so, it is convenient to assume that the points of $f(\vec{x})$ are arranged in descending order of amplitude (i.e. $f_1 \leq f_2 \leq \ldots \leq f_{M_f}$), and that $f(\vec{x})$ and $h(\vec{x})$ are scaled such that $f_1 = h_1 = 1$. Consideration of the mechanism of ghost formation by the shift-and-add algorithm (Hunt *et al.*, 1983) confirms that

$$h_m \leq f_m \tag{7.24}$$

where the equality holds for $m = 1$ (or, if some number M_e of points in $f(\vec{x})$ all share the largest amplitude, the equality holds for $m = 1, 2, \ldots, M_e$). When f_m is significantly less than unity, h_m tends to be significantly less than f_m. As an example of this characteristic of $h(\vec{x})$, a shift-and-add computer simulation by Davey (1989) of various two-point objects (i.e. $M_f = 2$), for which $f_2 = 1.00, 0.80, 0.60, 0.40$, resulted in values of $h_2 = 1.00, 0.30, 0.16, 0.07$ respectively.

By substituting (7.22) and (7.23) into (7.21), $g(\vec{x})$ is given by

$$g(\vec{x}) = \sum_{m=1}^{M_f} \sum_{n=1}^{M_f} f_m \, h_n \, \delta(\vec{x} - \vec{\alpha}_m + \vec{\alpha}_n) + c(\vec{x}) \tag{7.25}$$

Comparison of (7.25) with (3.25) reveals that $g(\vec{x})$ possesses points at identical locations to those in $a(\vec{x})$ (the autocorrelation of $f(\vec{x})$). This suggests that the autocorrelation tri-intersection method of §3.2.3.2 may be directly applicable to the problem of blindly deconvolving ghosted shift-and-add images. A triple product formed in the manner described in §3.2.3.2, but where $a(\vec{x})$ is replaced by $g(\vec{x})$, is here termed a *convolution triple product*, and denoted by $g^{(3)}(\vec{x})$. Thus $g^{(3)}(\vec{x})$ is given by

$$g^{(3)}(\vec{x}) = g(\vec{x})\, g(\vec{x} - \vec{s}_1)\, g(\vec{x} - \vec{s}_2) \tag{7.26}$$

Study of §3.2.3.2, with $a(\vec{x})$ replaced by $g(\vec{x})$, reveals that $g^{(3)}(\vec{x})$ must contain a translate of either $f(\vec{x})$ of $h(\vec{x})$ (provided that $f(\vec{x})$ satisfies the nonredundancy condition (3.33)). This is the basis for the proposed blind deconvolution method for shift-and-add images.

Such a technique for the blind deconvolution of shift-and-add images is clearly preferable to the general method for blind deconvolution of impulsive images (§7.2), since the selection of appropriate shift vectors is considerably simplified. For general impulsive images, the shift vectors can be determined from the autocorrelation of the convolution. For shift-and-add images, however, the shift vectors can be determined directly from the convolution, by exploiting the relationship between $f(\vec{x})$ and $h(\vec{x})$. Note that, because of this special relationship, $f(\vec{x})$ and $h(\vec{x})$ clearly violate (7.14). Thus a convolution triple product, at least, is required to effect the blind deconvolution of such images, rather than a convolution product, as is sufficient for general impulsive images (§7.2).

Determining the locations of points of the components of a convolution given by (7.25) is discussed in §7.3.1. The subsequent recovery of the amplitudes of the points is described in §7.3.2.

7.3.1 Determining Point Locations

An algorithm is now described which computes estimates of the components of a convolution, where it is known *a priori* that the components are of the form (7.22) and (7.23). Although the algorithm is based on the autocorrelation tri-intersection method of §3.2.3.2, it is implemented in a similar manner to Algorithm 7.1 in §7.2.1. Before detailing the algorithm's steps, mention is made of some of the ways in which the autocorrelation tri-intersection method is modified for the present purpose.

In the following discussion, $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ denote either estimates of $f(\vec{x})$ and $h(\vec{x})$, or estimates of $h(\vec{x})$ and $f(\vec{x})$, respectively. It is usually straightforward to determine whether $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$ corresponds to $f(\vec{x})$, by consideration of the ghost formation mechanism. Note that the major concern here is the determination of the locations of the points of each component. Thus, although one would hope that $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ have points at the correct locations (i.e. locations corresponding to those of the corresponding points in $f(\vec{x})$ and $h(\vec{x})$), it is of no concern if the points have incorrect amplitudes. The symbols $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ are later used to denote improved estimates of $f(\vec{x})$ and $h(\vec{x})$ (i.e. estimates with more accurate amplitudes). The subsequent determination of the point amplitudes is a more straightforward problem, and is discussed in §7.3.2.

In practice, any given convolution is necessarily contaminated. As the contamination level of $g(\vec{x})$ increases, so too does the number of points with significant value in the convolution triple product $g^{(3)}(\vec{x})$, as computed by (7.26). One way of reducing the number of points in $g^{(3)}(\vec{x})$ is to threshold $g(\vec{x})$, at a level tl equal to the estimated contamination level, before forming $g^{(3)}(\vec{x})$. By doing so, $g^{(3)}(\vec{x})$ only contains all M_f points of $f(\vec{x})$ if the amplitudes of all M_f points in each of the three translates of $f(\vec{x})$ contained in $g(\vec{x})$ which are superimposed in $g^{(3)}(\vec{x})$ exceed tl. Thus it is clear that the shift vectors $\vec{s}_1$ and $\vec{s}_2$ should be chosen so that the largest translates of $f(\vec{x})$ contained in $g(\vec{x})$ are superimposed in $g^{(3)}(\vec{x})$.

A simple strategy for selecting the shift vectors is now described. Firstly, $\vec{s}_1$ is chosen to be the vector from the origin to the point in $g(\vec{x})$ with largest amplitude (except that at $\vec{x} = \vec{0}$). Secondly, $\vec{s}_2$ is chosen to be the vector from the origin to the point in $g^{(2)}(\vec{x})$ with largest amplitude (except those points at $\vec{x} = \vec{0}, \vec{s}_1$ and $-\vec{s}_1$). Now it follows, from consideration of (7.25) and (7.24), that this strategy results in the shift vectors

$$\vec{s}_1 = \vec{\alpha}_2 - \vec{\alpha}_1 \quad \text{and} \quad \vec{s}_2 = \vec{\alpha}_3 - \vec{\alpha}_1 \tag{7.27}$$

Substituting (7.27) and (7.25) into (7.26), and assuming that $c(\vec{x}) = 0$ and that $f(\vec{x})$ satisfies (3.33), yields

$$g^{(3)}(\vec{x}) = h_1\, h_2\, h_3 \sum_{m=1}^{M_f} f_m^3\, \delta(\vec{x} - \vec{\alpha}_m + \vec{\alpha}_1) + \text{other terms} \tag{7.28}$$

where the other terms merely add to the amplitudes of the points at $\vec{x} = \vec{0}, \vec{s}_1$ and $\vec{s}_2$. The significance of (7.28) is that, when applying the above strategy to choose $\vec{s}_1$ and $\vec{s}_2$, the resulting $g^{(3)}(\vec{x})$ is formed by the superposition of the three largest translates of $f(\vec{x})$ contained in $g(\vec{x})$. This is precisely as desired for optimum noise tolerance.

When reconstructing an image-form from an autocorrelation, by the autocorrelation tri-intersection method, one is only concerned with obtaining an estimate of either $f(\vec{x})$ or $f(-\vec{x})$. In contrast, when solving a blind deconvolution problem, it is desirable to recover estimates of both $f(\vec{x})$ and $h(\vec{x})$, denoted by $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$, to allow comparison between $g(\vec{x})$ and $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$. Because of the special relationship between $f(\vec{x})$ and $h(\vec{x})$ (i.e. (7.22) and (7.23)), this can be achieved, after calculating $\tilde{f}(\vec{x}) = g^{(3)}(\vec{x})$, by simply setting $\tilde{h}(\vec{x}) = \tilde{f}(-\vec{x})$ (recall that the points in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ need not have the correct amplitudes).

However, calculating $\tilde{h}(\vec{x})$ in this manner can be ineffectual if $g^{(3)}(\vec{x})$ happens to contain $h(\vec{x})$, rather than $f(\vec{x})$. This can occur if the first few f_m have similar values, and $g(\vec{x})$ is contaminated. In such cases, a $g^{(3)}(\vec{x})$ which contains $h(\vec{x})$ often possesses significantly fewer points than a $g^{(3)}(\vec{x})$ which contains $f(\vec{x})$, due to the relative values, for each m, of f_m compared with h_m, especially as m approaches M_f. In other words, the three largest translates of $f(\vec{x})$ contained in $g(\vec{x})$ usually possess more points that are above tl than do the three largest translates of $h(\vec{x})$ contained in $g(\vec{x})$. Thus simply setting $\tilde{h}(\vec{x}) = \tilde{f}(-\vec{x})$, when $\tilde{f}(\vec{x})$ happens to contain $h(\vec{x})$, often results in $\tilde{h}(\vec{x})$ containing fewer points than $f(\vec{x})$.

Instead, it is more reliable to reconstruct $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ simultaneously. Consider a convolution triple product, formed with shift vectors $-\vec{s}_1$ and $-\vec{s}_2$, rather than $\vec{s}_1$ and $\vec{s}_2$, and denoted by $g_{-}^{(3)}(\vec{x})$. It follows, from (7.26), (7.25), and (7.27), that

$g_{-}^{(3)}(\vec{x})$ is given by

$$g_{-}^{(3)}(\vec{x}) = g(\vec{x})\, g(\vec{s}+\vec{s}_1)\, g(\vec{s}+\vec{s}_2) \tag{7.29a}$$

$$= f_1\, f_2\, f_3 \sum_{m=1}^{M_h} h_m^3\, \delta(\vec{x}+\vec{\alpha}_m-\vec{\alpha}_1) + \text{other terms} \tag{7.29b}$$

Thus $g_{-}^{(3)}(\vec{x})$ contains a translate of $h(\vec{x})$, and so $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ can be formed by (7.26) and (7.29a) simultaneously, employing the same shift vectors. The algorithm is now described:

Algorithm 7.3

Step 1: Form binary mask $g_{\mathrm{b}}(\vec{x}) = \begin{cases} 1, & \vec{x} : g(\vec{x}) \geq tl \\ 0, & \vec{x} : g(\vec{x}) < tl \end{cases}$.

Step 2: Form shift mask $mask(\vec{x}) = \begin{cases} 1, & \vec{x} : \vec{x} \neq 0 \\ 0, & \vec{x} : \vec{x} = 0 \end{cases}$.

Step 3: Set $\tilde{f}_1(\vec{x}) = g_{\mathrm{b}}(\vec{x})\, g(\vec{x})$.

Step 4: Set $\tilde{h}_1(\vec{x}) = \tilde{f}_1(\vec{x})$.

Step 5: Repeat the following steps, for $j = 2$ to J:

- Step 5.1: Find $\vec{s}_j$, the value of $\vec{x}$ at which $mask(\vec{x})\, \tilde{f}_{j-1}(\vec{x})$ is maximum.
- Step 5.2: Set $mask(\vec{s}_j) = 0$, and set $mask(-\vec{s}_j) = 0$.
- Step 5.3: Set $\tilde{f}_j(\vec{x}) = g_b(\vec{x} - \vec{s}_j)\, \tilde{f}_{j-1}(\vec{x})$.
- Step 5.4: Set $\tilde{h}_j(\vec{x}) = g_b(\vec{x} + \vec{s}_j)\, \tilde{h}_{j-1}(\vec{x})$.

Step 6: Set $\tilde{f}(\vec{x}) = \tilde{f}_J(\vec{x})$.

Step 7: Set $\tilde{h}(\vec{x}) = \tilde{h}_J(\vec{x})$.

There are significant aspects of Algorithm 7.3 which may not be immediately apparent from the above sequence of steps. The intermediate image estimates formed by combining j translates of $g(\vec{x})$ are denoted by $\tilde{f}_j(\vec{x})$ and $\tilde{h}_j(\vec{x})$, the final estimates (equivalent to $\tilde{f}_J$ and $\tilde{h}_J$) being denoted simply by $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$.

The binary mask $g_{\mathrm{b}}(\vec{x})$ serves the same purpose as that in Algorithm 7.1 of §7.2.1, as explained in the paragraph following Algorithm 7.1. Thus $\tilde{f}_3(\vec{x})$ is equivalent to $[g^{(3)}(\vec{x})]^{1/3}$ (apart from the amplitudes of the points at $\vec{x} = \vec{s}_1$ and $\vec{s}_2$). The shift mask, $mask(\vec{x})$, is employed to implement the selection of the shift vectors. Inspection of Algorithm 7.3 reveals that $\tilde{f}_{j-1}(\vec{x})$ corresponds to thresholded versions of $g(\vec{x})$ and $g^{(2)}(\vec{x})$ for $j = 2$ and $j = 3$ respectively, and that $mask(\vec{x})$ is zero at $\vec{x} = \vec{0}$ for $j = 2$ and zero at $\vec{x} = \vec{0}$, $\vec{s}_1$, and $-\vec{s}_1$ for $j = 3$. So it is evident that the strategy for the selection of shift vectors, as described in the paragraph containing (7.27), is indeed implemented by choosing $\vec{s}_j$ to be the vector from the origin to the largest pixel in $mask(\vec{x})\, \tilde{f}_{k-1}(\vec{x})$.

If $f(\vec{x})$ satisfies the nonredundancy condition (3.33), it follows that, for $J = 3$, $\tilde{f}(\vec{x})$ consists solely of a translate of $f(\vec{x})$, and $\tilde{h}(\vec{x})$ consists solely of a translate of $h(\vec{x})$, but both $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ have incorrect amplitudes at $\vec{x} = \vec{0}$. If $f(\vec{x})$ violates the nonredundancy condition, or if $g(\vec{x})$ is appreciably contaminated, spurious points may appear in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$. Such spurious points can usually be eliminated by combining more than three translates of $g(\vec{x})$ (i.e. choosing $K > 3$).

The recovery of the amplitudes of all the points of $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ is discussed in the following section.

7.3.2 Determining Point Amplitudes

The image estimates $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$, as generated by Algorithm 7.3, possess the correct number of points, and also the correct amplitude at all of the points (apart from those at $\vec{x} = \vec{0}$), provided that $f(\vec{x})$ satisfies condition (3.33), and $g(\vec{x})$ is uncontaminated. Significant contamination of $g(\vec{x})$, or violation of (3.33) by $f(\vec{x})$, may result in spurious points appearing in $\tilde{f}(\vec{x})$ and/or $\tilde{h}(\vec{x})$. Violation of (3.33) may also result in the amplitudes of some points in $\tilde{f}(\vec{x})$ and/or $\tilde{h}(\vec{x})$ (as well as those at $\vec{x} = \vec{0}$) being in error, if any point in $g(\vec{x})$ (apart from that at $\vec{x} = \vec{0}$) actually consists of two or more overlapped points. This subsection discusses means of computing improved estimates of $f(\vec{x})$ and $h(\vec{x})$, denoted by $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$, which have points at the same locations as those in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$, but (hopefully) have more accurate amplitudes.

It is now convenient to define $\tilde{g}(\vec{x}) = \tilde{f}(\vec{x}) \odot \tilde{h}(\vec{x})$, and to denote the number of points in $\tilde{f}(\vec{x})$, $\tilde{h}(\vec{x})$, and $\tilde{g}(\vec{x})$ by $M_{\tilde{f}}$, $M_{\tilde{h}}$, and $M_{\tilde{g}}$ respectively. Note that $M_{\tilde{g}}$ is, at most, $M_{\tilde{f}} M_{\tilde{h}} - \min(M_{\tilde{f}}, M_{\tilde{h}}) + 1$. By expressing $\hat{f}(\vec{x})$ as the sum of $M_{\tilde{f}}$ delta functions, the m^{th} of which has weight $\hat{f}_m$, and expressing $\hat{h}(\vec{x})$ similarly, determination of the amplitudes of all points of $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ reduces to finding the values of $\{\hat{f}_m : m = 1, \ldots, M_{\tilde{f}}\}$ and $\{\hat{h}_m : m = 1, \ldots, M_{\tilde{h}}\}$. By equating the amplitude of each point in $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$ to that of the corresponding point in $g(\vec{x})$, one obtains $M_{\tilde{g}}$ non-linear equations in the $M_{\tilde{f}} + M_{\tilde{h}}$ unknowns (i.e. all the $\hat{f}_m$ and $\hat{h}_m$). Since most of these equations involve only the product of one $\hat{f}_m$ with one $\hat{h}_m$, the system of equations is usually, in theory, significantly over-determined. In practice, no exact solution exists, due to the inevitable contamination of $g(\vec{x})$, but it should always be possible to find an approximate solution. In doing so, any points in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ with incorrect amplitude are corrected. Also, note that any spurious points in $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$ are eliminated, since the amplitude of any such point must evaluate to zero (or a small value, when $g(\vec{x})$ is contaminated).

For the sake of simplicity, it is now assumed that the amplitudes of all points in $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ are correct, apart from those at $\vec{x} = \vec{0}$. This assumption is only invalid if the nonredundancy condition (3.33) is violated, which is unlikely to occur if $f(\vec{x})$ comprises randomly positioned points, unless the number of points is large, or the spacing between points is small compared to the sample spacing. Correcting the amplitude of the point in $\tilde{f}(\vec{x})$ at $\vec{x} = \vec{0}$ can be effected straightforwardly, by appropriately combining two translates of $\tilde{f}(\vec{x})$ contained in $g(\vec{x})$; $\tilde{h}(\vec{x})$ can be similarly corrected. The algorithm's steps are:

Algorithm 7.4

Step 1: Find $\vec{\alpha}_2$, the value of $\vec{x}$ (except $\vec{0}$) at which $\tilde{f}(\vec{x})$ is maximum.

Step 2: Find $\vec{\alpha}_3$, the value of $\vec{x}$ (except $\vec{0}$ or $\vec{\alpha}_2$) at which $\tilde{f}(\vec{x})$ is maximum.

Step 3: Set $\hat{f}(\vec{x}) = \tilde{f}(\vec{x})$.

Step 4: Set $\hat{f}(\vec{0}) = \tilde{f}(\vec{\alpha}_3)\, g(-\vec{\alpha}_2)/g(\vec{\alpha}_3 - \vec{\alpha}_2)$.

Step 5: Repeat steps 1–4, exchanging $\tilde{f}(\vec{x}), \hat{f}(\vec{x})$ for $\tilde{h}(\vec{x}), \hat{h}(\vec{x})$.

To understand the operation of this algorithm, consider the expression (7.25) for $g(\vec{x})$, recalling that the f_m and h_m are arranged in descending order of amplitude. The problem now is to find f_1 and h_1, where $\vec{\alpha}_1 = \vec{0}$, given that $\{\hat{f}_m : m = 2, \ldots, M_{\hat{f}}\}$ and $\{\hat{h}_m : m = 2, \ldots, M_{\hat{h}}\}$ are known. Substituting various values of m and n into (7.25) yields

$$g(-\vec{\alpha}_2) = f_1 h_2 \quad \text{and} \quad g(\vec{\alpha}_3 - \vec{\alpha}_2) = f_3 h_2 \tag{7.30}$$

The ratio of $g(-\vec{\alpha}_2)$ to $g(\vec{\alpha}_3 - \vec{\alpha}_2)$ thus gives f_1/f_3, thereby allowing f_1 to be determined, since f_3 is simply given by $\tilde{f}(\vec{\alpha}_3)$. Furthermore, $\vec{\alpha}_2$ and $\vec{\alpha}_3$ can be immediately determined, being respectively the vector from the origin to the largest, and second largest, point in $g(\vec{x})$. The particular choice of m and n from which (7.30) is obtained is made to ensure the best performance when $g(\vec{x})$ is contaminated, by using the points of largest amplitude in $g(\vec{x})$, thus avoiding taking the ratio of small numbers.

Computational examples of Algorithms 7.3 and 7.4 are presented in the next subsection.

7.3.3 Computational Examples

The performance of the algorithms described in §§7.3.1 and 7.3.2, for blindly deconvolving shift-and-add images, is now demonstrated with shift-and-add images generated from computer simulated speckle images (see §2.8.2.2 for simulation details). The original image $f(\vec{x})$, shown in Figure 7.3a, comprises 10 points that are pseudo-randomly positioned in a 32×32 pixel frame, with amplitudes pseudo-randomly distributed in the range [0.1,1]. The defogged shift-and-add image $g(\vec{x})$, generated from $f(\vec{x})$ with N_{frames} (i.e. the total number of speckle images employed) set to $N_{\text{frames}} = 400$, is shown in Figure 7.3b, while the corresponding deconvolved image estimates $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ are shown in Figure 7.3c and d respectively. Algorithm 7.4 was employed to compute $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$, from $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ as computed by Algorithm 7.3, with $J = 3$ and $tl = 0.02$ (where $g(\vec{x})$ is normalized such that $\max\{g(\vec{x})\} = 1$. Note the similarity between $\hat{f}(\vec{x})$ and $f(\vec{x})$. In practice, of course, one cannot make such a comparison, but it is possible to compare $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$ with $g(\vec{x})$. Figure 7.3e shows $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$ (displayed on the same grey-scale as $g(\vec{x})$), which is encouragingly similar to $g(\vec{x})$, and so one could be confident that $\hat{f}(\vec{x})$ is a useful estimate of $f(\vec{x})$.

For the example presented in Figure 7.3, the threshold level was chosen to be $tl = 0.02$. This value was selected following inspection of the histogram of $g(\vec{x})$, which

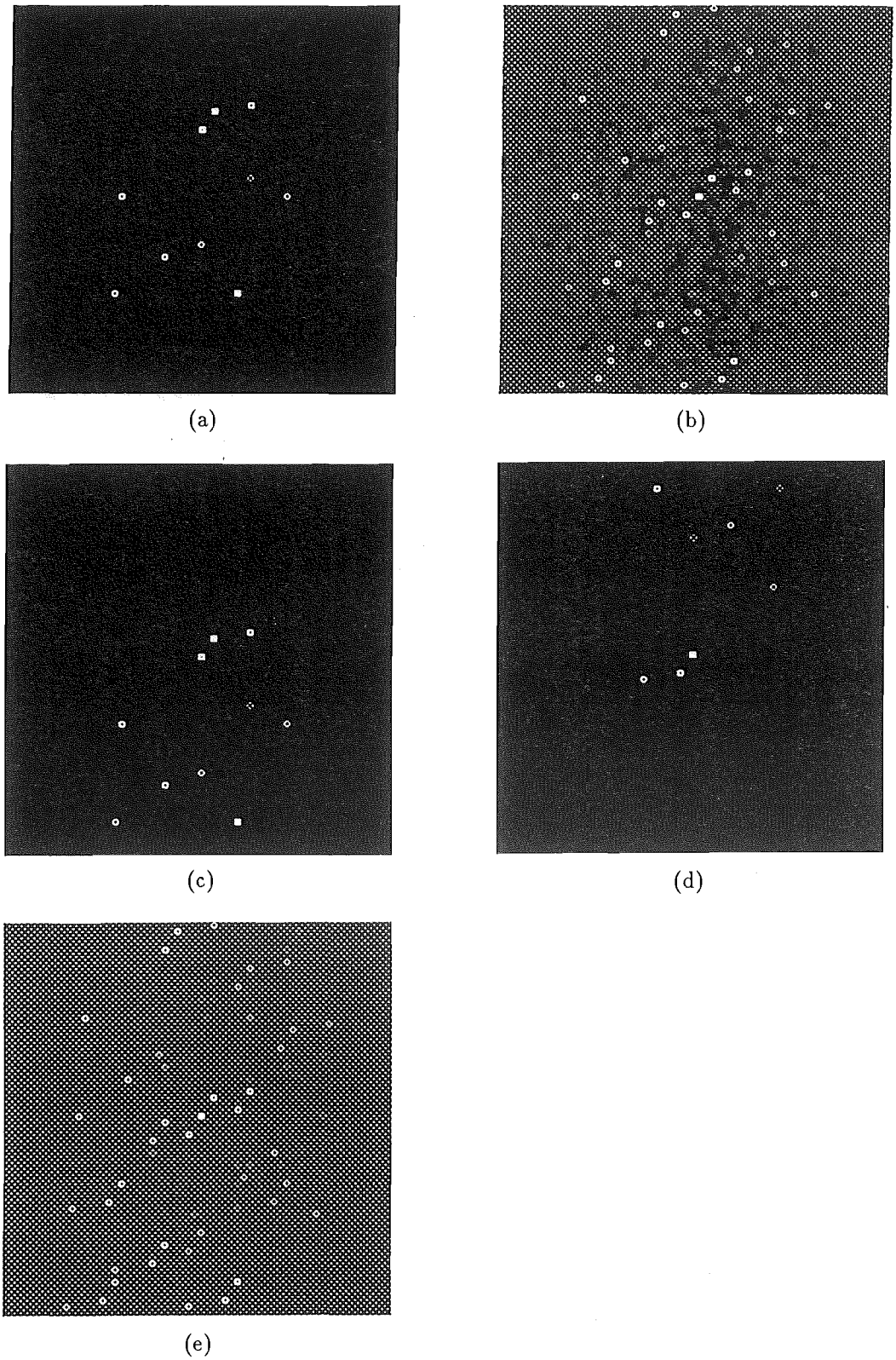


Figure 7.3 Illustration of blind deconvolution of ghosted shift-and-add image (Algorithms 7.3 and 7.4). (a) original image $f(\vec{x})$, (b) defogged shift-and-add image $g(\vec{x})$ ($N_{\text{frames}} = 400$), (c) image estimate $\hat{f}(\vec{x})$, (d) psf estimate $\hat{h}(\vec{x})$, and (e) reconvolved estimates $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$. Note that in both (b) and (e), the pixel at the origin has been set to 1.1 times the value of the next largest pixel.

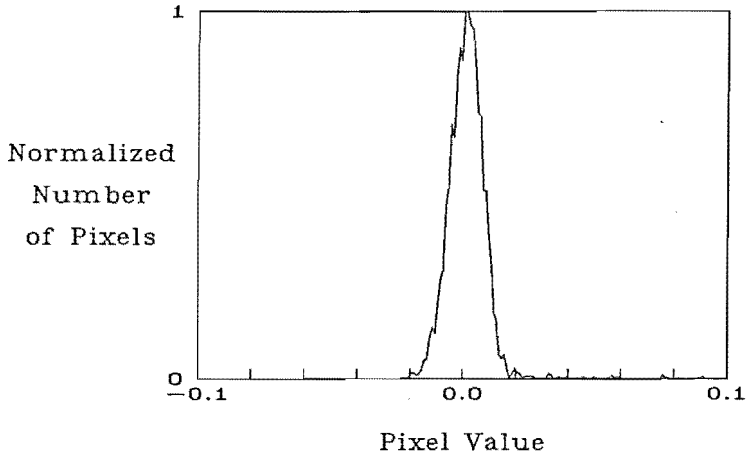


Figure 7.4 Histogram of pixel values of $g(\vec{x})$ in Figure 7.3b.

is shown in Figure 7.4. Note that $g(\vec{x})$ contains 76 pixels with amplitude greater than tl. It is worthwhile discussing the behaviour of Algorithm 7.3 with different values of tl. If too large a value is chosen for tl, some points in $f(\vec{x})$ or $h(\vec{x})$ may not be present in $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$. This occurs if the amplitude of any point in any of the J brightest translates of $f(\vec{x})$ or $h(\vec{x})$ that are contained in $g(\vec{x})$ is below tl. If, on the other hand, too small a value is chosen for tl, spurious points may appear in $\tilde{f}(\vec{x})$ and/or $\tilde{h}(\vec{x})$. This occurs, for $J = 3$, if a point in $g_b(\vec{x})$ coincides with a point in both $g_b(\vec{x} - \vec{s}_1)$ and $g_b(\vec{x} - \vec{s}_2)$, where at least one of these points is not due to a point of $f(\vec{x}) \odot h(\vec{x})$ (i.e. that point is due to contamination, with amplitude greater than tl, of a pixel in $g(\vec{x})$). As mentioned in §7.3.1, spurious points can often be eliminated by choosing $J > 3$. However, it is desirable to choose the smallest possible J, since all points in the J brightest translates of $f(\vec{x})$ in $g(\vec{x})$ must have amplitude greater than tl to ensure that either $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$ contains the same number of points as $f(\vec{x})$.

The notation $NP\{q\}$ is now invoked to denote the number of points in an image $q(\vec{x})$. Figure 7.5 shows the variation of $NP\{\tilde{f}\}$, $NP\{\tilde{h}\}$, and $NP\{g_b\}$ with tl, as computed by Algorithm 7.3 operating with $J = 3$ upon the $g(\vec{x})$ shown in Figure 7.3b. Note that as the value of tl is decreased, $NP\{g_b\}$ increases steadily, whereas $NP\{\tilde{f}\}$ initially increases, then remains constant over a significant range of values of tl, before again increasing. This behaviour suggests how an automatic strategy could be devised for choosing the optimum tl for any given contaminated $g(\vec{x})$. For example, one could monitor $NP\{\tilde{f}\}$ as tl is decreased, and note when a significant increase in $NP\{\tilde{f}\}$ occurs. The optimum value of tl is then taken to be that in effect just before the aforesaid increase in $NP\{\tilde{f}\}$.

The results of applying Algorithm 7.3 (with $J = 3$ and $tl = 0.015$) and Algorithm 7.4 to another shift-and-add image, generated from 400 simulated speckle images, are presented in Figure 7.6.

The advantage of choosing $J > 3$ in Algorithm 7.3, when the level of contamination on $g(\vec{x})$ is appreciable compared with tl, is now demonstrated. Figure 7.7b shows the defogged shift-and-add image $g(\vec{x})$, generated from $f(\vec{x})$ shown in Figure 7.7a with $N_{\text{frames}} = 100$. The image estimates $\hat{f}(\vec{x})$, computed by Algorithms 7.3 and 7.4 (with $tl = 0.015$), are shown in Figures 7.7c, d, and e for $J = 3$, 4, and 5

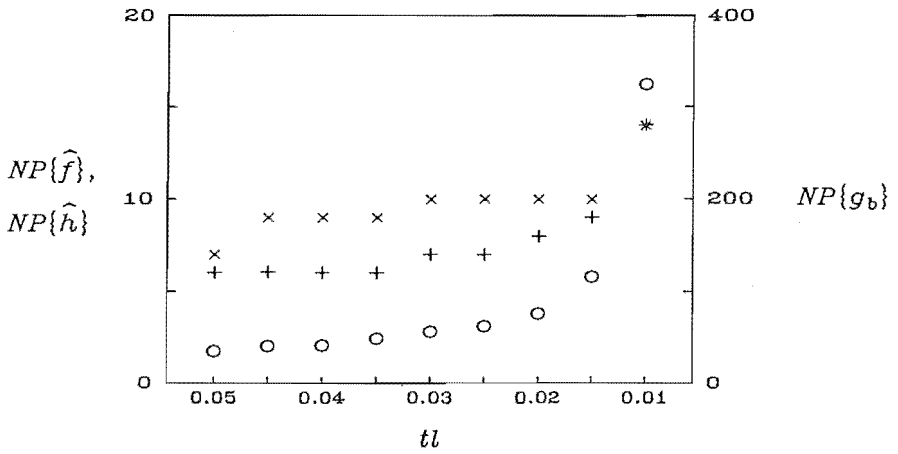


Figure 7.5 Dependence on threshold level of number of points in $\hat{f}(\vec{x})$, $\hat{h}(\vec{x})$, and $g_b(\vec{x})$, for $g(\vec{x})$ shown in Figure 7.3b. $NP\{\tilde{f}\} = \times$, $NP\{\tilde{h}\} = +$, $NP\{\tilde{g}_b\} = \circ$.

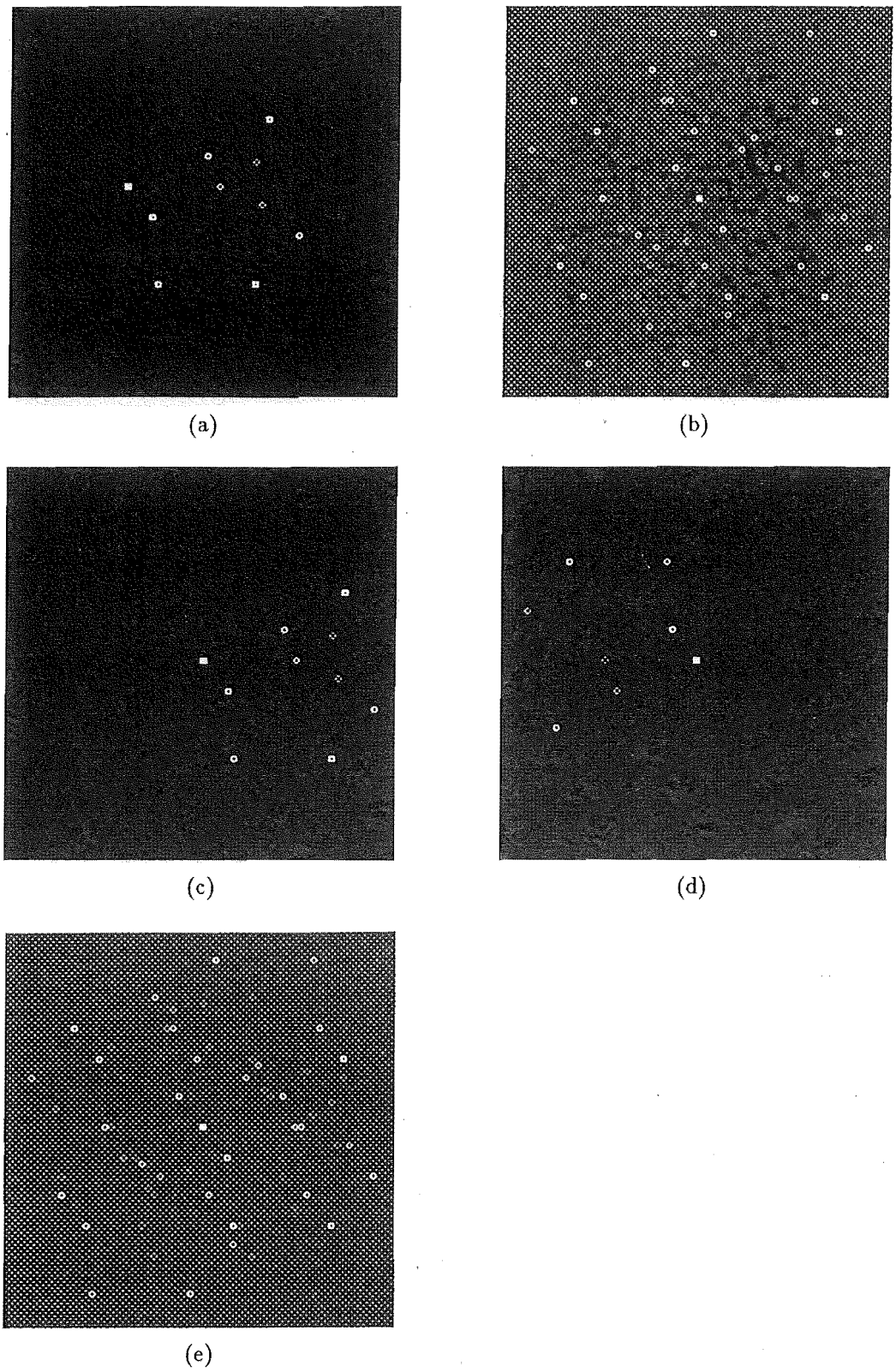
respectively. Note the multitude of spurious points in $\hat{f}(\vec{x})$ for $J = 3$, all of which are eliminated by choosing $J = 5$.

7.4 Discussion

It has been indicated in §7.1 that the incorporation of an impulsiveness constraint into the iterative blind deconvolution algorithm is difficult, and may indeed be detrimental to the algorithm's convergence. A new technique to effect the blind deconvolution of impulsive images, by combining appropriate translates of the convolution, has been described in §7.2. For situations in which the components of the given convolution are unrelated, it has been shown that effective shift vectors can be determined from the autocorrelation of the convolution.

A practical situation where it is desired to blindly deconvolve the convolution of two impulsive images arises with ghosted shift-and-add images. In this case, the components of the convolution are closely related, a fact that is exploited by the techniques described in §7.3, which are here referred to collectively as the *convolution tri-intersection method.* In this method, the shift vectors are obtained directly from the convolution, in a manner similar to the autocorrelation tri-intersection method, and both convolution components are recovered simultaneously. This method appears to be effective when operating on simulated shift-and-add data, despite significant contamination.

It is now worthwhile reiterating the situations in which $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ (as computed by Algorithms 7.3 and 7.4) should be useful estimates of $f(\vec{x})$ and $h(\vec{x})$. If the contamination in $g(\vec{x})$ is small (compared with the amplitude of the smallest point in $f(\vec{x}) \odot h(\vec{x})$), then $\hat{f}(\vec{x})$ ought to be a useful estimate of $f(\vec{x})$. However, appreciable contamination in $g(\vec{x})$ can result in spurious points appearing in $\hat{f}(\vec{x})$. Choosing $J > 3$ in Algorithm 7.3 can often remove such points, but may also remove some of the desired points. In such cases, forming $\tilde{f}(\vec{x})$ and $\tilde{h}(\vec{x})$ from Algorithm 7.3 with $J = 3$, then solving the set of equations for the amplitudes of the points in $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ (as described in §7.3.2) may prove to be the most effective strategy for



(a) (b) (c) (d) (e)

Figure 7.6 Second illustration of blind deconvolution of ghosted shift-and-add image (Algorithms 7.3 and 7.4). (a) original image $f(\vec{x})$, (b) defogged shift-and-add image $g(\vec{x})$ ($N_{\text{frames}} = 400$), (c) image estimate $\hat{f}(\vec{x})$, (d) psf estimate $\hat{h}(\vec{x})$, (e) reconvolved estimates $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$. Note that, in both (b) and (e), the pixel at the origin has been set to 1.1 times the value of the next largest pixel, as in Figure 7.3.

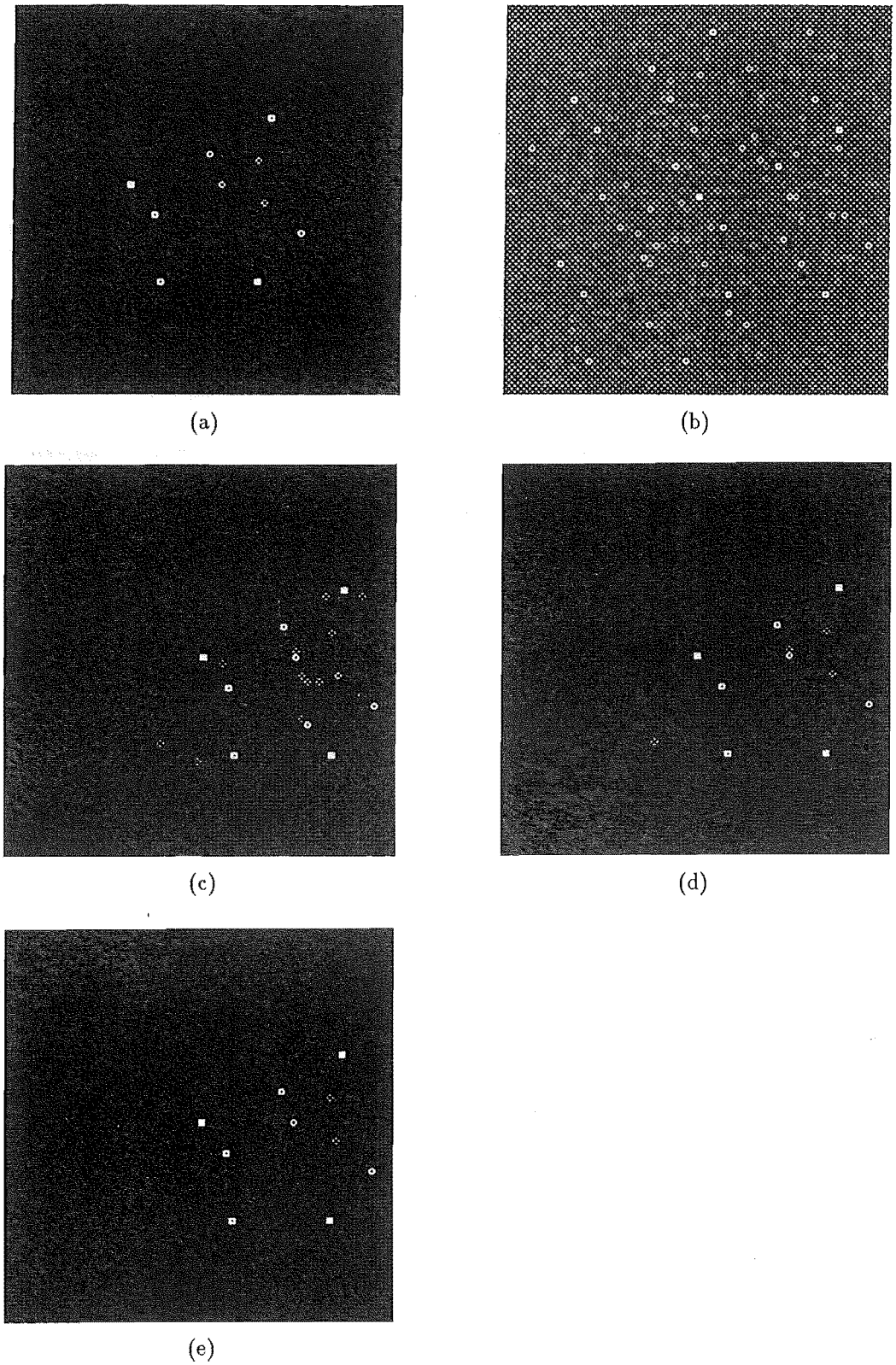


Figure 7.7 Effect of the value of J in Algorithm 7.3. (a) original image $f(\vec{x})$, (b) defogged shift-and-add image $g(\vec{x})$ ($N_{\text{frames}} = 100$), and image estimates $\hat{f}(\vec{x})$ for (c) $J = 3$, (d) $J = 4$, and (e) $J = 5$.

removing spurious points. Alternatively, spurious points in $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ could be detected (and then corrected) by comparing $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$ with $g(\vec{x})$, since a spurious point in $\hat{f}(\vec{x})$ at $\vec{x}'$ gives rise to an extra $\hat{h}(\vec{x} - \vec{x}')$ in $\hat{f}(\vec{x}) \odot \hat{h}(\vec{x})$ which does not appear in $g(\vec{x})$. Furthermore, the precise nature of the points in any measured $g(\vec{x})$ must be taken into account, when determining criteria for deciding which points are indeed overlapped in a convolution triple product.

However, when considering any further enhancements of the convolution tri-intersection method, alternative techniques for the blind deconvolution of shift-and-add images should be borne in mind, such as the iterative blind deconvolution algorithm. The relative merits of these two methods of deghosting shift-and-add images of point objects are now briefly mentioned. The convolution tri-intersection method is significantly less computationally demanding than the iterative blind deconvolution algorithm, since the latter requires repeated FFTs. Furthermore, the convolution tri-intersection method implicitly applies an impulsiveness constraint to the recovered images, whereas it is difficult to do so with the iterative blind deconvolution algorithm. However, the major disadvantage of the convolution tri-intersection method appears to be that spurious points may arise in the image estimates if the convolution is significantly contaminated, and efforts to remove such spurious points may also remove some desired points.

An appropriate combination of the convolution tri-intersection method and the iterative blind deconvolution algorithm may eventually prove to be an effective strategy for deghosting shift-and-add images of point objects. One such possibility is to use the images produced by the convolution tri-intersection method as starting images for the iterative blind deconvolution algorithm, which ought to significantly speed its convergence.

Another possibility, which is likely to be more effective, is to use the images produced by the convolution tri-intersection method to form image masks for use by the iterative blind deconvolution algorithm. This is simply achieved by forming $S\{f(\vec{x})\}$ which is everywhere zero, except at values of $\vec{x}$ where $\tilde{f}(\vec{x})$ is nonzero, at which points its value is 1. In a similar fashion, $S\{h(\vec{x})\}$ is formed, and these masks are applied in Steps (3.83b) and (3.83d) of the iterative blind deconvolution algorithm (§3.3.3.2). The iterative blind deconvolution algorithm is thereby used to solve for the amplitudes of the points of $\hat{f}(\vec{x})$ and $\hat{h}(\vec{x})$ (as mentioned in §7.3.2). Note that it is of little consequence if $\tilde{f}(\vec{x})$ or $\tilde{h}(\vec{x})$ contain spurious points, since the values recovered by the iterative blind deconvolution algorithm at these points should be small. Furthermore, the set of images with support $S\{f(\vec{x})\}$ is convex, and so the convergence of the iterative blind deconvolution algorithm should not be impaired by the incorporation of such a constraint. Indeed, since this support possesses such a specialized shape, one would expect the convergence of the algorithm to be significantly faster than when employing a simple rectangular support.

Chapter 8

Conclusions and Suggestions for Further Research

In this final chapter, conclusions regarding the original research presented in Chapters 4 to 7 are drawn. Furthermore, a number of promising avenues for further research into various aspects of phase retrieval and blind deconvolution are discussed. The conclusions are presented in §8.1, whilst the suggestions for further research are made in §8.2.

8.1 Conclusions

Detailed conclusions concerning the original work presented in Chapters 4, 5, 6, and 7 are included in the discussions in §§4.6, 5.2.4,5.3.3,5.4.3, 6.4, and 7.4 respectively. A brief summary of these conclusions is now presented. Conclusions relating to the work on phase retrieval are given in §8.1.1, with those relating to the work on blind deconvolution being given in §8.1.2.

8.1.1 Phase Retrieval

The phase problem is introduced in §2.1, and existing methods for effecting its solution are reviewed in §3.2. A number of new phase retrieval techniques are presented in Chapters 4 and 5, some of which augment existing phase retrieval algorithms, while others constitute original phase retrieval algorithms. The previously presented conclusions concerning each of these techniques are now summarized.

In Chapter 4, it is demonstrated that the hybrid input-output algorithm behaves erratically when operating on appreciably contaminated Fourier magnitude data. The image error is shown to exhibit significant fluctuations, with those reconstructed image-forms corresponding to local minima in the error curve tending to be the most faithful. A new algorithm is described which selects a number of such image-forms, appropriately averaging them to produce an enhanced image-form. This algorithm is shown to produce enhanced image-forms that are significantly more faithful than the image-forms produced by the standard hybrid input-output algorithm, when operating on contaminated Fourier magnitude data.

It is pointed out in Chapter 5 that all currently available phase retrieval

algorithms are computationally intensive. Three new phase retrieval strategies are described, each of which attempts to reduce the computational expenditure that is required to effect phase retrieval in various differing situations. The conclusions concerning each of these strategies are now summarized.

In §5.2, it is demonstrated that the hybrid input-output algorithm tends to initially recover the gross features of the reconstructed image. A new phase retrieval strategy is described which operates on increasing numbers of Fourier magnitude samples as the iterations proceed. This strategy is shown to be significantly less computationally demanding, when reconstructing a positive image exhibiting intricate detail, than the standard hybrid input-output algorithm.

It is reasoned in §5.3 that the hybrid input-output algorithm may not perform satisfactorily in situations where positivity is the sole image space constraint that can be enforced. A new input-output algorithm is proposed, which comprises an appropriate combination of the basic and hybrid input-output algorithms. It is demonstrated that the performance of this modified input-output algorithm, for an example in which only a positivity constraint is employed, is superior to that of the standard hybrid input-output algorithm.

A new method of invoking simulated annealing to effect phase retrieval, termed the Fourier space simulated annealing algorithm, is described in §5.4. Its effectiveness in recovering the phase of a small complex image is demonstrated. Since it appears to be difficult to incorporate a positivity constraint into the Fourier space simulated annealing algorithm, it seems that this algorithm is unlikely to rival the image-space simulated annealing algorithm when recovering positive images. However, when recovering bipolar or complex images, it is argued that the Fourier space simulated annealing algorithm may be significantly less computationally demanding than the image-space simulated annealing algorithm. Since this algorithm is computationally intensive, it may prove to be most effective when appropriately combined with the hybrid input-output algorithm. Alternatively, it may be useful in situations where the hybrid input-output algorithm tends to be ineffective.

8.1.2 Blind Deconvolution

The blind deconvolution problem is introduced in §2.6, and existing methods for effecting its solution are reviewed in §3.3. Two new blind deconvolution techniques are introduced in Chapters 6 and 7. The previously presented conclusions regarding each of these techniques are now summarized.

A new blind deconvolution technique is detailed in Chapter 6, which appropriately invokes the simulated annealing algorithm. It is demonstrated that this technique, termed the simulated annealing blind deconvolution algorithm, is capable of effecting the blind deconvolution of contaminated convolution data. However, this algorithm is computationally intensive, and so may prove to be most effective when used in conjunction with the iterative blind deconvolution algorithm. Incorporating a regularization constraint into the simulated annealing blind deconvolution algorithm holds promise of producing a robust algorithm. Such an algorithm may well prove to be an effective means of obtaining useful solutions, or improving solutions obtained by the iterative blind deconvolution algorithm, to blind deconvolution problems in

which the convolution data are appreciably contaminated.

Two new blind deconvolution algorithms, for blindly deconvolving the convolution of impulsive images, are described in Chapter 7. In contrast to iterative blind deconvolution algorithms, these direct algorithms operate by appropriately combining a number of translates of the given convolution. The first algorithm is applicable to convolutions of two impulsive components which bear no special relationship to one another. The second algorithm is only applicable in situations where it is known *a priori* that the locations of the points of one component are the same as those of the other component, apart from being rotated by 180°. It is demonstrated that the second algorithm is an effective means of deghosting shift-and-add images that are formed from computer-generated speckle images.

8.2 Suggestions for Further Research

Many suggestions are made, in the discussions in §§4.6, 5.2.4, 5.3.3, 5.4.3, 6.4, and 7.4, as to how each of the new techniques described in Chapters 4 to 7 may be improved. There is no point in repeating these suggestions here. However, it is worthwhile mentioning some other promising avenues for further research into various aspects of the phase problem and blind deconvolution problem. Suggestions for further research into the phase problem are given in §8.2.1, and those for the blind deconvolution problem are given in §8.2.2.

8.2.1 Phase Retrieval

The hybrid input-output algorithm, as described in §3.2.8.2, is generally the most effective phase retrieval algorithm that is presently available. A number of promising investigations, which relate to the hybrid input-output algorithm, are now suggested.

As mentioned in §5.3.1, the reasoning on which the hybrid input-output algorithm was originally based appears to be somewhat *ad hoc*. It is thus worthwhile to continue to seek alternative input-output algorithms. The value of doing so is confirmed by the favourable performance of the modified input-output algorithm, as demonstrated in §5.3.2.

It is suggested in §4.6 that an iterative phase retrieval algorithm which enforces the image space constraints at each iteration may have an advantage over one which enforces the Fourier space constraints at each iteration (as is performed by the hybrid input-output algorithm). To devise such an algorithm, it is useful to consider the third, fourth, and first steps of the error-reduction algorithm, i.e. (3.49c,d,a), to be a non-linear system, as depicted in Figure 8.1. Note that this approach is in contrast to that described in §3.2.8.2, in which the first, second, and third steps of the error-reduction algorithm are considered as a non-linear system, as depicted in Figure 3.2. It is now conjectured that a change made to the input $\bar{F}_k[p,q]$ at some pixel $[p',q']$ is likely to result in a similar (although possibly scaled) change to the output $\tilde{F}_k[p,q]$ at $[p',q']$. This then allows various algorithms to be devised, which vary $\bar{F}_k[p,q]$ in such a manner as to "persuade" $\tilde{F}_k[p,q]$ to meet the Fourier space constraints.

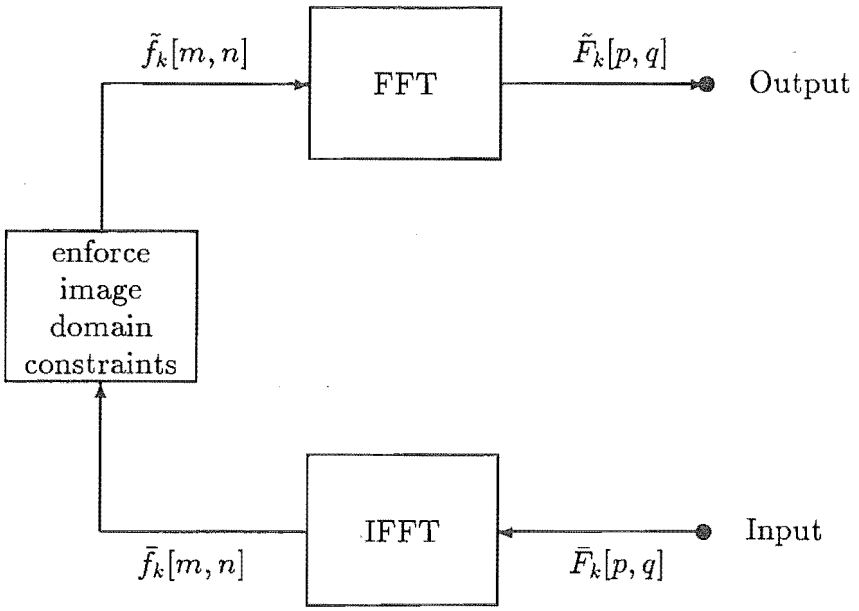


Figure 8.1 Depiction of the third, fourth, and first steps of the error-reduction algorithm (3.49c,d,a). By considering this process as a non-linear system, alternative input-output algorithms can be derived.

One such algorithm is characterized by computing the next input as

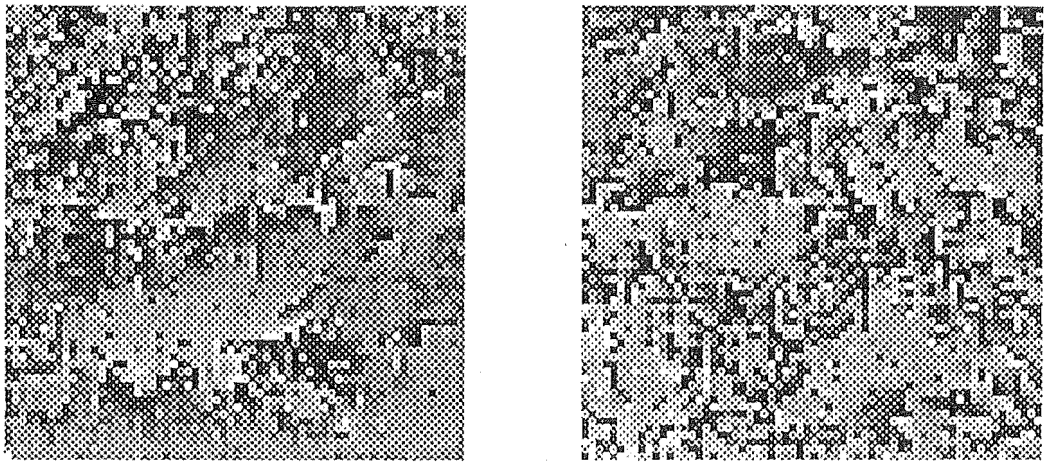
$$\bar{F}_{k+1}[p,q] = \left[|\bar{F}_k[p,q]| - \beta(|\tilde{F}_k[p,q]| - M[p,q])\right] \exp(i\,\mathrm{Ph}\{\tilde{F}_k[p,q]\}) \qquad (8.1)$$

where $M[p,q]$ denotes the available Fourier magnitude data, and β (a positive constant) is a gain parameter. The performance of this algorithm, and other input-output algorithms based upon the non-linear system depicted in Figure 8.1, should be investigated. Such algorithms could well prove to be superior to the standard hybrid input-output algorithm in some situations (such as when the data are appreciably contaminated), or may be useful in conjunction with the standard hybrid input-output algorithm.

It would be worthwhile to investigate whether the sampling theorem based phase retrieval algorithm of Arsenault and Chalasinska-Macukow (1983), as described in §3.2.5.1, is capable of converging to the desired solution when initiated with an image containing only the gross features of the original image. If it is capable of doing so, an appropriate combination of the hybrid input-output algorithm and the sampling theorem based method may prove to be a significantly less computationally demanding phase retrieval strategy than the use of the hybrid input-output algorithm alone.

A number of authors (e.g. Fienup, 1987) report that the convergence of the hybrid input-output algorithm, when a non-centrosymmetric image mask is employed, tends to be significantly more rapid then that when a centrosymmetric mask is employed. An explanation for this observation is now offered.

Recall that Figure 5.3 in §5.2.1 shows the reconstructed images $\hat{f}[m,n]$ that are obtained after various numbers of hybrid input-output iterations have been performed on the Fourier magnitude of the original image $f[m,n]$, the central 64×64 pixel portion of which is shown in Figure 5.1. It is now informative to examine the quan-



(a) (b)

Figure 8.2 Portions of (a) $(\mathrm{Ph}\{\hat{F}[p,q]\} - \mathrm{Ph}\{F[p,q]\}) \bmod 2\pi$ and (b) $(\mathrm{Ph}\{\hat{F}[p,q]\} + \mathrm{Ph}\{F[p,q]\}) \bmod 2\pi$, where $f[m,n]$ and $\hat{f}[m,n]$ are shown in Figure 5.1 and Figure 5.3a respectively.

tities $(\mathrm{Ph}\{\hat{F}[p,q]\} - \mathrm{Ph}\{F[p,q]\}) \bmod 2\pi$ and $(\mathrm{Ph}\{\hat{F}[p,q]\} + \mathrm{Ph}\{F[p,q]\}) \bmod 2\pi$. Figure 8.2 shows a 64×64 pixel portion (selected such that the lower right corner of each portion corresponds to the origin of Fourier space) of the aforementioned 128×128 pixel quantities, computed from $\hat{f}[m,n]$, the central 64×64 pixel portion of which is shown in Figure 5.3a. The regions of uniform brightness in Figure 8.2a correspond to regions of Fourier space in which $\mathrm{Ph}\{\hat{F}[p,q]\}$ is equivalent to (i.e. equal to, apart from a linear phase shift) $\mathrm{Ph}\{F[p,q]\}$. Conversely, the regions of uniform brightness in Figure 8.2b correspond to regions of Fourier space in which $\mathrm{Ph}\{\hat{F}[p,q]\}$ is equivalent to $\mathrm{Ph}\{F^*[p,q]\}$.

Thus it is apparent that the hybrid input-output algorithm is attempting to recover both $f[m,n]$ and $f^*[-m,-n]$ simultaneously, thereby resulting in slow convergence. It is worth considering whether such behaviour could somehow be avoided.

It should be noted that Fienup and Wackerman (1986) report that the hybrid input-output algorithm can stagnate when the reconstructed image $\hat{f}[m,n]$ contains both $f[m,n]$ and $f^*[-m,-n]$ in approximately equal proportions. However, the point being made in the above two paragraphs is that the convergence of the hybrid input-output algorithm appears to be impeded by $\hat{f}[m,n]$ containing both $f[m,n]$ and $f^*[-m,-n]$, even when either $f[m,n]$ or $f^*[-m,-n]$ is dominated by the other to such a degree that it is not visible in $\hat{f}[m,n]$.

There are numerous modifications that could be made to the image-space simulated annealing algorithm of Nieto-Vesperinas *et al.* (1988), described in §3.2.9.2. Some such possibilities are now outlined.

As mentioned in §6.4, it appears that varying α (the scale of perturbations) in proportion to either $\sqrt{T}$ or $\sqrt{Q}$ would be an effective strategy. Alternatively, adjusting α automatically, in the manner described in §6.3, could well be preferable. It may even be worth investigating employing a different value of α for each pixel of the image, determined automatically along the lines described by Corana *et al.* (1987).

As mentioned in §3.1.2.2, generalizations of the simulated annealing algo-

rithm can be devised by choosing acceptance probability functions other than the exponential function. It may be worth experimenting with such algorithms.

It would be well worth investigating the incorporation of a regularization constraint into the simulated annealing phase retrieval algorithm. This could be implemented in a similar manner to that suggested in §6.4, for incorporating a regularization constraint into the simulated annealing blind deconvolution algorithm. The resulting algorithm is likely to be robust in the presence of appreciable contamination on the Fourier magnitude data.

8.2.2 Blind Deconvolution

It is emphasized, in §2.6, that the phase problem is merely a special case of the blind deconvolution problem. Thus it is worthwhile to consider whether there are any phase retrieval techniques that can be usefully generalized, so as to apply to the blind deconvolution problem.

Since the iterative blind deconvolution algorithm (as described in §3.3.3.2) is the most promising blind deconvolution algorithm at present, some schemes which may improve its performance are now suggested.

The best performance of the iterative blind deconvolution algorithm results when image masks are employed which enclose the supports of the convolution's components as tightly as possible. It is thus worthwhile to investigate how well the supports of a convolution's components can be inferred from the support of the convolution itself. The work of Crimmins *et al.* (1990), on estimating the support of an object from the support of its autocorrelation, could form the basis for such an investigation.

If the iterative blind deconvolution algorithm is found to stagnate, it would be worthwhile attempting to incorporate an input-output strategy into the algorithm. However, this must be attempted with due care. The input-output strategy should be applied to a part of the iterative loop where the input-output philosophy, as described in §3.2.8.2, does indeed hold.

The deconvolution steps of the iterative blind deconvolution algorithm, i.e. (3.83a,c), are crucial to the effectiveness of the algorithm, and ought to be critically examined. The operation of the algorithm of Davey *et al.* (1989), as described in §3.3.3.2, is now considered. It is assumed for convenience that $\alpha = 0$ in (3.85). Suppose that the square of the magnitude of $\tilde{H}_k[p,q]$ is small, compared with Φ, at some pixel $[p',q']$ in Fourier space. The Wiener filter (3.85) then forces the magnitude of $\bar{F}_k[p,q]$ to also be small at $[p',q']$. If $|F[p,q]|$ happens to have significant value at $[p',q']$, then the algorithm may be prevented from converging to a satisfactory solution.

It is thus worthwhile to consider how $\bar{F}_k[p,q]$ may be determined at values of p and q for which $|\tilde{H}_k[p,q]|$ is sufficiently small that $G[p,q]/\tilde{H}_k[p,q]$ is not a useful estimate of $F[p,q]$. Provided that strong *a priori* constraints (such as support and/or positivity) are available regarding $f[m,n]$, it is reasonable to attempt to obtain $\bar{F}_k[p,q]$ at such values by constrained interpolation. The deconvolution scheme employed in the algorithm of Ayers and Dainty (1988), namely (3.84), does indeed allow such interpolation to occur. Interpolation of $\bar{F}_k[p,q]$ can be incorporated into the algorithm

of Davey *et al.* (1989) by adding an extra term to (3.85), giving

$$\bar{F}_k[p,q] = \frac{G[p,q]\,(\tilde{H}_k[p,q])^*}{|\tilde{H}_k[p,q]|^2 + \Phi} + \frac{\Phi\,\tilde{F}_k[p,q]}{|\tilde{H}_k[p,q]|^2 + \Phi} \tag{8.2}$$

where α has been set to zero in (3.85). It is convenient to express (8.2) as

$$\bar{F}_k[p,q] = \phi[p,q]\,\frac{G[p,q]}{\tilde{H}_k[p,q]} + (1 - \phi[p,q])\,\tilde{F}_k[p,q] \tag{8.3}$$

where

$$\phi[p,q] = \frac{|\tilde{H}_k[p,q]|^2}{|\tilde{H}_k[p,q]|^2 + \Phi} \tag{8.4}$$

At values of p and q for which $|\tilde{H}_k[p,q]|^2 \gg \Phi$, $\bar{F}_k[p,q]$ is simply $G[p,q]$ inverse-filtered by $\tilde{H}_k[p,q]$. Where $|\tilde{H}_k[p,q]|^2 \ll \Phi$, $\bar{F}_k[p,q]$ is merely $\tilde{F}_k[p,q]$, thereby allowing interpolation of $\tilde{F}_k[p,q]$ to occur. Furthermore, the transition between these two limiting cases is smooth, unlike that in (3.84), being conditioned by the value of Φ.

References

ABLES, J.G. (1974), 'Maximum entropy spectral analysis', *Astronomy and Astrophysics Supplement Series*, Vol. 15, No. 3, June, pp. 383–393.

ANDREWS, H.C. and HUNT, B.R. (1977), *Digital Image Restoration*, Prentice-Hall, New Jersey.

ANILY, S. and FEDERGRUEN, A. (1987), 'Simulated annealing methods with general acceptance probabilities', *Journal of Applied Probability*, Vol. 24, No. 3, September, pp. 657–667.

ARSENAULT, H.H. and CHALASINSKA-MACUKOW, K. (1983), 'The solution to the phase retrieval problem using the sampling theorem', *Optics Communications*, Vol. 47, No. 6, October, pp. 380–386.

AYERS, G.R. and DAINTY, J.C. (1988), 'An iterative blind deconvolution algorithm and its applications', *Optics Letters*, Vol. 13, No. 7, July, pp. 547–549.

BAGNUOLO, JR., W.G. (1982), 'The application of Bates' algorithm to binary stars', *Monthly Notices of the Royal Astronomical Society*, Vol. 200, No. 3, September, pp. 1113–1122.

BAGNUOLO, JR., W.G. (1985), 'Image restoration by the shift-and-add algorithm', *Optics Letters*, Vol. 10, No. 5, May, pp. 200–202.

BALDWIN, J.E. and WARNER, P.J. (1978), 'Phaseless aperture synthesis', *Monthly Notices of the Royal Astronomical Society*, Vol. 182, No. 2, February, pp. 411–422.

BARAKAT, R. and NEWSAM, G. (1984), 'Necessary conditions for a unique solution to two-dimensional phase recovery', *Journal of Mathematical Physics*, Vol. 25, No. 11, November, pp. 3190–3193.

BAREKET, N. and KOLIOPOULOS, C.L. (Eds.) (1982), *Wavefront Sensing*, Proceedings SPIE Vol. 351.

BATES, R.H.T. (1969), 'Contributions to the theory of intensity interferometry', *Monthly Notices of the Royal Astronomical Society*, Vol. 142, No. 4, pp. 413–428.

BATES, R.H.T. (1982a), 'Astronomical speckle imaging', *Physics Reports*, Vol. 90, pp. 203–297.

BATES, R.H.T. (1982b), 'Fourier phase problems are uniquely solvable in more than one dimension. I: Underlying theory', *Optik*, Vol. 61, No. 3, June, pp. 247–262.

BATES, R.H.T. (1984), 'Uniqueness of solution to two-dimensional Fourier phase problems for localised and positive images', *Computer Vision, Graphics and Image Processing*, Vol. 25, No. 2, February, pp. 205–217.

BATES, R.H.T. and CADY, F.M. (1980), 'Towards true imaging by wideband speckle interferometry', *Optics Communications*, Vol. 32, No. 3, March, pp. 365–369.

BATES, R.H.T. and DAVEY, B.L.K. (1987a), 'Towards making shift-and-add a versatile imaging technique', in IDELL, P.S. (Ed.), *Digital Image Recovery and Synthesis*, Proceedings SPIE Vol. 828, pp. 87–94.

BATES, R.H.T. and DAVEY, B.L.K. (1987b), 'Computationally-cost-effective speckle imaging', in *Proceedings of the Joint Workshop on High-Resolution Imaging from the Ground Using Interferometric Techniques*, Oracle, Arizona, January 12–15, pp. 219–222.

BATES, R.H.T. and FRIGHT, W.R. (1983), 'Composite two-dimensional phase restoration procedure', *Journal of the Optical Society of America*, Vol. 73, No. 3, March, pp. 358–365.

BATES, R.H.T. and FRIGHT, W.R. (1984), 'Reconstructing images from their Fourier intensities', in HUANG, T.S. (Ed.), *Advances in Computer Vision and Image Processing: Vol. 1 — Image Reconstruction from Incomplete Observations*, JAI Press Inc., Greenwich, Connecticut, Chap. 5, pp. 227–264.

BATES, R.H.T. and GOUGH, P.T. (1975), 'New outlook on processing radiation received from objects viewed through randomly fluctuating media', *IEEE Transactions on Computers*, Vol. C-24, No. 4, April, pp. 449–456.

BATES, R.H.T. and LANE, R.G. (1987), 'Automatic deconvolution and phase retrieval', in IDELL, P.S. (Ed.), *Digital Image Recovery and Synthesis*, Proceedings SPIE Vol. 828, pp. 158–164.

BATES, R.H.T. and McDONNELL, M.J. (1986), *Image Restoration and Reconstruction*, Clarendon Press, Oxford.

BATES, R.H.T. and MNYAMA, D. (1986), 'The status of practical Fourier phase retrieval', in HAWKES, P.W. (Ed.), *Advances in Electronics and Electron Physics, Vol. 67*, Academic Press, pp. 1–64.

BATES, R.H.T. and TAN, D.G.H. (1986), 'Phase retrieval for electron diffraction patterns suggesting orientational order without translational symmetry', *Optik*, Vol. 73, No. 2, May, pp. 69–73.

BATES, R.H.T., NAPIER, P.J., McKINNON, A.E. and McDONNELL, M.J. (1976), 'Self consistent deconvolution. I: Theory', *Optik*, Vol. 44, No. 2, January, pp. 183–201.

BATES, J.H.T., FRIGHT, W.R. and BATES, R.H.T. (1984), 'Wiener filtering and cleaning in a general image processing context', *Monthly Notices of the Royal Astronomical Society*, Vol. 211, No. 1, November, pp. 1–14.

BATES, R.H.T., SINTON, A.M. and MINARD, R.A. (1985), 'Generalization of shift-and-add imaging', in ARSENAULT, H.H. (Ed.), *International Conference on Speckle*, Proceedings SPIE Vol. 556, August 20-23, pp. 263–269.

BATES, R.H.T., QUEK, B.K. and PARKER, C.R. (1990), 'Some implications of zero-sheets for blind deconvolution and phase retrieval', Submitted to: *Journal of the Optical Society of America A.*

BERENYI, H.M., DEIGHTON, H.V. and FIDDY, M.A. (1985), 'The use of bivariate polynomial factorization algorithms in two-dimensional phase problems', *Optica Acta*, Vol. 32, No. 6, June, pp. 689–701.

BOHACHEVSKY, I.O., JOHNSON, M.E. and STEIN, M.L. (1986), 'Generalized simulated annealing for function optimization', *Technometrics*, Vol. 28, No. 3, August, pp. 209–217.

BRACEWELL, R.N. (1978), *The Fourier Transform and its Applications*, McGraw-Hill, New York.

BRAMES, B.J. (1987), 'Testing for support irreducibility', *Journal of the Optical Society of America A*, Vol. 4, No. 1, pp. 135–147.

BRICOGNE, G. (1984), 'Maximum entropy and the foundations of direct methods', *Acta Crystallographica*, Vol. A40, No. 4, July, pp. 410–455.

BRIGHAM, E.O. (1974), *The Fast Fourier Transform*, Prentice-Hall.

BRUCK, Y.M. and SODIN, L.G. (1979), 'On the ambiguity of the image reconstruction problem', *Optics Communications*, Vol. 30, No. 3, September, pp. 304–308.

BRÜNGER, A.T., KURIYAN, J. and KARPLUS, M. (1987), 'Crystallographic r-factor refinement by molecular dynamics', *Science*, Vol. 235, pp. 458–460.

BRYAN, R.K. and SKILLING, J. (1986), 'Maximum entropy image reconstruction from phaseless Fourier data', *Optica Acta*, Vol. 33, No. 3, March, pp. 287–299.

BUERGER, M.J. (1944), *The Photography of the Reciprocal Lattice*, The American Society for X-ray and Electron Diffraction.

BURCH, S.F., GULL, S.F. and SKILLING, J. (1983), 'Image restoration by a powerful maximum entropy method', *Computer Vision, Graphics, and Image Processing*, Vol. 23, No. 2, August, pp. 113–128.

BURGE, R.E., FIDDY, M.A., GREENAWAY, A.H. and ROSS, G. (1976), 'The phase problem', *Proceedings Royal Society of London A*, Vol. 350, No. 1661, 20th August, pp. 191–212.

CADY, F.M. (1980), *Applications of Microcomputers in Interactive Image Processing*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

CADZOW, J.A. and SUN, Y. (1986), 'Sequences with positive semidefinite Fourier transforms', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 6, December, pp. 1502–1510.

CANTERAKIS, N. (1983), 'Magnitude-only reconstruction of two-dimensional sequences with finite regions of support', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, No. 5, October, pp. 1256–1262.

ÇETIN, A.E. and ANSARI, R. (1988), 'Convolution-based framework for signal recovery and applications', *Journal of the Optical Society of America A*, Vol. 5, No. 8, August, pp. 1193–1200.

CEDERQUIST, J.N., FIENUP, J.R., MARRON, J.C. and PAXMAN, R.G. (1988), 'Phase retrieval from experimental far-field speckle data', *Optics Letters*, Vol. 13, No. 8, August, pp. 619–621.

CHALASINSKA-MACUKOW, K. and ARSENAULT, H.H. (1985), 'Fast iterative solution to exact equations for the two-dimensional phase-retrieval problem', *Journal of the Optical Society of America A*, Vol. 2, No. 1, January, pp. 46–50.

CHRISTIANSEN, W.N. and HÖGBOM, J.A. (1985), *Radiotelescopes*, Cambridge University Press, Cambridge, UK, 2 ed.

CHRISTOU, J. (1981), 'Imaging of star clusters from speckle interferometry', *Optics Communications*, Vol. 37, No. 5, 1st June, pp. 331–334.

CHURCHILL, R.H., BROWN, J.W. and VERHEY, R.F. (1974), *Complex Variables and Applications*, McGraw Hill.

COOLEY, J.W. and TUKEY, J.W. (1965), 'An algorithm for the machine calculation of complex Fourier series', *Mathematics of Computation*, Vol. 19, No. 90, April, pp. 297–301.

CORANA, A., MARCHESI, M. and RIDELLA, S. (1987), 'Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm', *ACM Transactions on Mathematical Software*, Vol. 13, No. 3, September, pp. 262–280.

COWLEY, J. and SMITH, D. (1987), 'The present and future of high-resolution electron microscopy', *Acta Crystallographica*, Vol. A43, No. 6, November, pp. 737–751.

CRIMMINS, T.R. (1987), 'Phase retrieval for discrete functions with support constraints', *Journal of the Optical Society of America A*, Vol. 4, No. 1, January, pp. 124–134.

CRIMMINS, T.R. and FIENUP, J.R. (1981), 'Ambiguity of phase retrieval for functions with disconnected support', *Journal of the Optical Society of America*, Vol. 71, No. 8, August, pp. 1026–1028.

CRIMMINS, T.R. and FIENUP, J.R. (1983), 'Uniqueness of phase retrieval for functions with sufficiently disconnected support', *Journal of the Optical Society of America*, Vol. 73, No. 2, February, pp. 218–221.

CRIMMINS, T.R., FIENUP, J.R. and THELEN, B.J. (1990), 'Improved bounds on object support from autocorrelation support and application to phase retrieval', *Journal of the Optical Society of America A*, Vol. 7, No. 1, January, pp. 3–13.

DAINTY, J.C. (1973), 'Diffraction-limited imaging of stellar objects using telescopes of low optical quality', *Optics Communications*, Vol. 7, No. 2, February, pp. 129–134.

DAINTY, J.C. (Ed.) (1975), *Laser Speckle and Related Phenomena*, Springer-Verlag, Berlin, 1st ed.

DAINTY, J.C. and FIENUP, J.R. (1987), 'Phase retrieval and image reconstruction for astronomy', in STARK, H. (Ed.), *Image Recovery : Theory and Application*, Academic Press, Orlando, Florida, Chap. 7, pp. 231–275.

DALLAS, W.J. (1976), 'Digital computation of image complex-amplitude from intensities in two image-space planes', *Optics Communications*, Vol. 18, No. 3, August, pp. 317–320.

DAVEY, B.L.K. (1989), *Advances in Blind Deconvolution*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

DAVEY, B.L.K., LANE, R.G. and BATES, R.H.T. (1989), 'Blind deconvolution of noisy complex-valued image', *Optics Communications*, Vol. 69, No. 5–6, 15th January, pp. 353–356.

DEIGHTON, H.V., SCIVIER, M.S. and FIDDY, M.A. (1985), 'Solution of the two-dimensional phase retrieval problem', *Optics Letters*, Vol. 10, No. 6, June, pp. 250–251.

DENNIS, J.E. and SCNABEL, R.B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, New Jersey.

FELDKAMP, G.B. and FIENUP, J.R. (1980), 'Noise properties of images reconstructed from Fourier modulus', in RHODES, W.T. (Ed.), *1980 International Optical Computing Conference*, Proceedings SPIE Vol. 231, pp. 84–93.

FIDDY, M.A., BRAMES, B.J. and DAINTY, J.C. (1983), 'Enforcing irreducibility for phase retrieval in two dimensions', *Optics Letters*, Vol. 8, pp. 96–98.

FIENUP, J.R. (1978), 'Reconstruction of an object from the modulus of its Fourier transform', *Optics Letters*, Vol. 3, No. 1, July, pp. 27–29.

FIENUP, J.R. (1980), 'Iterative method applied to image recovery and to computer-generated holograms', *Optical Engineering*, Vol. 19, No. 3, May/June, pp. 297–305.

FIENUP, J.R. (1981a), 'Autocorrelation unfolding', in RHODES, W.T., FIENUP, J.R. and SALEH, B.E.A. (Eds.), *Transformations in Optical Signal Processing*, Proceedings SPIE Vol. 373, pp. 203–209.

FIENUP, J.R. (1981b), 'Reconstruction and synthesis applications of an iterative algorithm', in RHODES, W.T., FIENUP, J.R. and SALEH, B.E.A. (Eds.), *Transformations in Optical Signal Processing*, Proceedings SPIE Vol. 373, pp. 147–160.

FIENUP, J.R. (1982), 'Phase retrieval algorithms: A comparison', *Applied Optics*, Vol. 21, No. 15, 1st August, pp. 2758–2769.

FIENUP, J.R. (1983a), 'Reconstruction of objects having latent reference points', *Journal of the Optical Society of America*, Vol. 73, No. 11, November, pp. 1421–1426.

FIENUP, J.R. (1983b), 'Experimental evidence of the uniqueness of phase retrieval from intensity data', in ROBERTS, J.A. (Ed.), *Indirect Imaging*, Cambridge University Press, pp. 99–109.

FIENUP, J.R. (1987), 'Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint', *Journal of the Optical Society of America A*, Vol. 4, No. 1, January, pp. 118–123.

FIENUP, J.R. and WACKERMAN, C.C. (1986), 'Phase-retrieval stagnation problems and solutions', *Journal of the Optical Society of America A*, Vol. 3, No. 11, November, pp. 1897–1907.

FIENUP, J.R., CRIMMINS, T.R. and HOLSZTYNSKI, W. (1982), 'Reconstruction of the support of an object from the support of its autocorrelation', *Journal of the Optical Society of America*, Vol. 72, No. 5, May, pp. 610–624.

FRIED, D.L. (1966), 'Optical resolution through a randomly inhomogeneous medium for very long and very short exposures', *Journal of the Optical Society of America*, Vol. 56, No. 10, October, pp. 1372–1379.

FRIGHT, W.R. (1984), *The Fourier Phase Problem*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

FRIGHT, W.R. and BATES, R.H.T. (1982), 'Fourier phase problems are uniquely solvable in more than one dimension. III: Computational examples for two dimensions', *Optik*, Vol. 62, No. 3, October, pp. 219–230.

GALAGHER, N. and LIU, B. (1973), 'Method for computing kinoforms that reduces image reconstruction errors', *Applied Optics*, Vol. 12, No. 10, October, pp. 2328–2335.

GARDENIER, P.H. (1990), *Antenna Aperture Phase Retrieval*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

GARDENIER, P.H., MCCALLUM, B.C. and BATES, R.H.T. (1986a), 'Fourier transform magnitudes are unique pattern recognition templates', *Biological Cybernetics*, Vol. 54, pp. 385–391.

GARDENIER, P.H., LIM, C.A., TAN, D.G.H. and BATES, R.H.T. (1986b), 'Aperture distribution phase from single radiation pattern measurement via Gerchberg-Saxton algorithm', *Electronics Letters*, Vol. 22, No. 2, January, pp. 113–115.

GEMAN, S. and GEMAN, D. (1984), 'Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, November, pp. 721–741.

GERCHBERG, R.W. (1974), 'Super-resolution through error energy reduction', *Optica Acta*, Vol. 21, No. 9, September, pp. 709–720.

GERCHBERG, R.W. (1986), 'The lock problem in the Gerchberg-Saxton algorithm for phase retrieval', *Optik*, Vol. 74, No. 3, October, pp. 91–93.

GERCHBERG, R.W. (1989), 'Superresolution through error function extrapolation', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 10, October, pp. 1603–1606.

GERCHBERG, R.W. and SAXTON, W.O. (1972), 'A practical algorithm for the determination of phase from image and diffraction plane pictures', *Optik*, Vol. 35, No. 2, April, pp. 237–246.

GONSALVES, R.A. (1982), 'Fundamentals of wavefront sensing by phase retrieval', in BAREKET, N. and KOLIOPOULOS, C.L. (Eds.), *Wavefront Sensing*, Proceedings SPIE Vol. 351, pp. 56–65.

GOODMAN, J.W. (1968), *Fourier Optics*, McGraw-Hill, New York.

GREENAWAY, A.H. (1977), 'Proposal for phase recovery from a single intensity distribution', *Optics Letters*, Vol. 1, No. 1, July, pp. 10–12.

GULL, S.F. and DANIELL, G.J. (1978), 'Image reconstruction from incomplete and noisy data', *Nature*, Vol. 272, pp. 686–690.

GULL, S.F. and SKILLING, J. (1984), 'Maximum entropy method in image processing', *Proceedings IEE Part F*, Vol. 131, No. 6, October, pp. 646–659.

HAMMERSLEY, J.M. and HANDSCOMB, D.C. (1964), *Monte Carlo Methods*, Methuen & Co., London.

HARRIS, F.J. (1978), 'On the use of windows for harmonic analysis with the discrete Fourier transform', *IEEE Proceedings*, Vol. 66, No. 1, January, pp. 51–83.

HAWKES, P.W. (1972), *Electron Optics and Electron Microscopy*, Taylor and Francis, London.

HAYES, M.H. (1982), 'The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 2, April, pp. 140–154.

HAYES, M.H. and McCLELLAN, J.H. (1982), 'Reducible polynomials in more than one variable', *IEEE Proceedings*, Vol. 70, No. 2, February, pp. 197–198.

HAYES, M.H., LIM, J.S. and OPPENHEIM, A.V. (1980), 'Signal reconstruction from phase or magnitude', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 6, December, pp. 672–680.

HILDEBRANDT, J.W. (1987), 'Blind deconvolution', *Mathematical Scientist*, Vol. 12, pp. 39–51.

HOFSTETTER, E.M. (1964), 'Construction of time limited functions with specified autocorrelation functions', *IEEE Transactions on Information Theory*, Vol. IT-10, pp. 119–126.

HÖGBOM, J.A. (1974), 'Aperture synthesis with a non-regular distribution of interferometer baselines', *Astronomy and Astrophysics Supplement Series*, Vol. 15, No. 3, June, pp. 417–426.

HUNT, B.R., FRIGHT, W.R. and BATES, R.H.T. (1983), 'Analysis of the shift-and-add method for imaging through turbulent media', *Journal of the Optical Society of America*, Vol. 73, No. 4, April, pp. 456–465.

IZRAELEVITZ, D. and LIM, J.S. (1987), 'A new direct algorithm for image reconstruction from Fourier transform magnitude', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 4, April, pp. 511–519.

KARLE, J. and HAUPTMAN, H. (1950), 'The phases and magnitudes of the structure factors', *Acta Crystallographica*, Vol. 3, No. 3, May, pp. 181–187.

KIRKPATRICK, S., GELATT, JR., C.D. and VECCHI, M.P. (1983), 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, pp. 671–680.

KNOX, K.T. and THOMPSON, B.J. (1974), 'Recovery of images from atmospherically degraded short-exposure photographs', *The Astrophysical Journal Letters*, Vol. 193, No. 1, 1st October, pp. L45–L48.

LABEYRIE, A. (1970), 'Attainment of diffraction-limited resolution in large telescopes by Fourier analysing speckle patterns in star images', *Astronomy and Astrophysics*, Vol. 6, No. 1, May, pp. 85–87.

LANE, R.G. (1987), 'Recovery of complex images from Fourier magnitude', *Optics Communications*, Vol. 63, No. 1, 1st July, pp. 6–10.

LANE, R.G. (1988), *Blind Deconvolution and Phase Retrieval*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand, March.

LANE, R.G. and BATES, R.H.T. (1987a), 'Automatic multi-dimensional deconvolution', *Journal of the Optical Society of America A*, Vol. 4, No. 1, January, pp. 180–188.

LANE, R.G. and BATES, R.H.T. (1987b), 'Relevance for blind deconvolution of recovering Fourier magnitude', *Optics Communications*, Vol. 63, No. 1, 1 July, pp. 11–14.

LANE, R.G., FRIGHT, W.R. and BATES, R.H.T. (1987), 'Direct phase retrieval', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 4, April, pp. 520–526.

LAWTON, W. (1981), 'Uniqueness results for the phase retrieval problem for radial functions', *Journal of the Optical Society of America*, Vol. 71, No. 12, December, pp. 1519–1522.

LEVI, A. and STARK, H. (1983), 'Signal restoration from phase by projections onto convex sets', *Journal of the Optical Society of America*, Vol. 73, No. 6, June, pp. 810–822.

LEVI, A. and STARK, H. (1984), 'Image restoration by the method of generalized projections with application to restoration from magnitude', *Journal of the Optical Society of America A*, Vol. 1, No. 9, September, pp. 932–943.

LEVI, A. and STARK, H. (1987), 'Restoration from phase and magnitude by generalised projections', in STARK, H. (Ed.), *Image Recovery : Theory and Application*, Academic Press, Orlando, Florida, Chap. 8, pp. 277–320.

LIPSON, H. and TAYLOR, C.A. (1958), *Fourier Transforms and X-ray Diffraction*, G. Bell and sons, London.

LYNDS, C.R., WORDEN, S.P. and HARVEY, J.W. (1976), 'Digital image reconstruction applied to Alpha Orionis', *The Astrophysical Journal*, Vol. 207, No. 1, 1 July, pp. 174–180. Plates 3, 4.

MARKS, R.J. and SMITH, M.J. (1981), 'Closed-form object restoration from limited spatial and spectral information', *Optics Letters*, Vol. 6, No. 11, November, pp. 522–524.

MCCALLUM, B.C. (1990), 'Blind deconvolution by simulated annealing', *Optics Communications*, Vol. 75, No. 2, pp. 101–105.

MCCALLUM, B.C. and BATES, R.H.T. (1989), 'Towards a strategy for automatic phase retrieval from noisy Fourier intensities', *Journal of Modern Optics*, Vol. 36, No. 5, May, pp. 619–648.

MCCALLUM, B.C., GARDENIER, P.H. and BATES, R.H.T. (1986), 'Invertible invariant transformations for robotic catalogues', in *Proceedings of the International Conference on Future Computing Systems*, pp. 151–158.

METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. and TELLER, E. (1953), 'Equation of state calculations by fast computing machines', *Journal of Chemical Physics*, Vol. 21, No. 4, June, pp. 1087–1092.

MINARD, R.A. (1985), *Imaging in a Distorting Medium*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

MISELL, D.L. (1973), 'An examination of an iterative method for the solution of the phase problem in optics and electron optics: I. Test calculations', *Journal of Physics D: Applied Physics*, Vol. 6, No. 18, 5th December, pp. 2200–2216.

MITRA, D., ROMEO, F. and SANGIOVANNI-VINCENTELLI, A. (1986), 'Convergence and finite-time behaviour of simulated annealing', *Advances in Applied Probability*, Vol. 18, pp. 747–771.

MNYAMA, D. (1987), *Image Reconstruction with Incomplete Data and Partial Constraints*, Ph.D. thesis, University of Canterbury, New Zealand.

NAKAJIMA, N. and ASAKURA, T. (1986), 'Two-dimensional phase retrieval using the logarithmic Hilbert transform and the estimation technique of zero information', *Journal of Physics D: Applied Physics*, Vol. 19, No. 3, 14th March, pp. 319–331.

NAPIER, P.J. (1971), *Reconstruction of Radiating Sources*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand.

NAPIER, P.J. and BATES, R.H.T. (1974), 'Inferring phase information from modulus information in two-dimensional aperture synthesis', *Astronomy and Astrophysics Supplement Series*, Vol. 15, No. 3, June, pp. 427–430.

NARAYAN, N. (1987), 'Phase retrieval with the maximum entropy method', in GOAD, J.W. (Ed.), *Proceedings of the Joint Workshop on High-Resolution Imaging from the Ground Using Interferometric Techniques*, National Optical Astronomy Observatories, Oracle, Arizona, January 12-15, pp. 183–186.

NARAYAN, R. and NITYANANDA, R. (1986), 'Maximum entropy image restoration in astronomy', *Annual Review of Astronomy and Astrophysics*, Vol. 24, pp. 127–170.

NARAYAN, R., NITYANANDA, R. and VANI, G.V. (1983), 'Maximum entropy – A new approach to the crystallographic phase problem', *Proceedings Indian Academy of Science (Chemical Science)*, Vol. 92, No. 4&5, pp. 341–358.

NAVARRO, R., FUENTES, F.J. and NIETO-VESPERINAS, M. (1989), 'Simulated annealing image reconstruction in photon-limited stellar speckle interferometry', *Astronomy and Astrophysics*, Vol. 208, No. 1/2, January(1), pp. 374–380.

NEWMAN, B.B. and HILDEBRANDT, J. (1987), 'Blind image restoration', *The Australian Computer Journal*, Vol. 19, No. 3, August, pp. 126–133.

NIETO-VESPERINAS, M. (1980), 'Dispersion relations in two dimensions: Application to the phase problem', *Optik*, Vol. 56, No. 4, September, pp. 377–384.

NIETO-VESPERINAS, M. (1986), 'A study of the performance of nonlinear least-square optimization methods in the problem of phase retrieval', *Optica Acta*, Vol. 33, No. 6, June, pp. 713–722.

NIETO-VESPERINAS, M. and DAINTY, J.C. (1985), 'A note on Eisenstein's irreducibility criterion for two-dimensional sampled objects', *Optics Communications*, Vol. 54, No. 6, 15th July, pp. 333–334.

NIETO-VESPERINAS, M. and DAINTY, J.C. (1986), 'Phase recovery for two-dimensional digital objects by polynomial factorisation', *Optics Communications*, Vol. 58, No. 2, 15 May, pp. 83–88.

NIETO-VESPERINAS, M. and MENDEZ, J.A. (1986), 'Phase retrieval by Monte-Carlo methods', *Optics Communications*, Vol. 59, No. 4, 15th September, pp. 249–254.

NIETO-VESPERINAS, M., NAVARRO, R. and FUENTES, F.J. (1988), 'Performance of a simulated-annealing algorithm for phase retrieval', *Journal of the Optical Society of America A*, Vol. 5, No. 1, January, pp. 30–38.

NITYANANDA, R. and NARAYAN, R. (1982), 'Maximum entropy image reconstruction - a practical non-information-theoretic approach', *Journal of Astrophysics and Astronomy*, Vol. 3, No. 4, pp. 419–450.

OPPENHEIM, A.V. and LIM, J.S. (1981), 'The importance of phase in signals', *IEEE Proceedings*, Vol. 69, pp. 529–541.

OPPENHEIM, A.V. and SCHAFER, R.W. (1975), *Digital Signal Processing*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

PAPOULIS, A. (1975), 'A new algorithm in spectral analysis and band-limited extrapolation', *IEEE Transactions on Circuits and Systems*, Vol. 22, No. 9, September, pp. 735–742.

PAPOULIS, A. (1980), *Circuits and Systems*, Holt, Rinehart and Winston.

PINCUS, M. (1968), 'A closed form solution of certain programming problems', *Operations Research*, Vol. 16, No. 3, May, pp. 690–694.

PINCUS, M. (1970), 'A Monte-Carlo method for the approximate solution of certain types of constrained optimization problems', *Operations Research*, Vol. 18, No. 6, May, pp. 1225–1229.

POLAK, E. (1971), *Computational Methods in Optimization*, Academic Press, New York.

PRESS, W.H., FLANNERY, B.P., TEUKOLSKY, S.A. and VETTERLING, W.T. (1986), *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK.

PRYCE, J.D. (1972), *Basic Methods of Linear Functional Analysis*, Hutchinson & Co., London.

QUATIERI, T.F. (1981), 'Recursive two-dimensional signal reconstruction from linear system input and output magnitudes', *IEEE Proceedings*, Vol. 69, No. 5, May, pp. 667–668.

RAMACHANDRAN, G.N. and SRINIVASAN, R. (1970), *Fourier Methods in Crystallography*, Wiley-Interscience, New York.

REQUICHA, A.A.G. (1980), 'The zeros of entire functions: Theory and engineering applications', *IEEE Proceedings*, Vol. 68, No. 3, March, pp. 308–328.

RIBAK, E., HEGE, E.K. and CHRISTOU, J.C. (1985), 'Use of matched filtering to identify speckle locations', in ARSENAULT, H.H. (Ed.), *International Conference on Speckle*, Proceedings SPIE Vol. 556, August 20-23, pp. 196–201.

RODDIER, F. (1981), 'The effects of atmospheric turbulence in optical astronomy', in WOLF, E. (Ed.), *Progress in Optics Vol. XIX*, North-Holland, pp. 283–376.

RODDIER, F. (1988), 'Interferometric imaging in optical astronomy', *Physics Reports*, Vol. 170, No. 2, November, pp. 97–166.

ROSS, G., FIDDY, M.A., NIETO-VESPERINAS, M. and WHEELER, M.W.L. (1978), 'The phase problem in scattering phenomena: the zeros of entire functions and their significance', *Proceedings Royal Society of London A*, Vol. 360, pp. 25–45.

SABRI, M.S. and STEENAART, W. (1978), 'An approach to band-limited extrapolation: the extrapolation matrix', *IEEE Transactions on Circuits and Systems*, Vol. CAS-25, No. 2, February, pp. 74–78.

SALEH, B.E.A. (1987), 'Image synthesis : discovery instead of recovery', in STARK, H. (Ed.), *Image Recovery : Theory and Application*, Academic Press, Orlando, Florida, Chap. 12, pp. 463–498.

SANZ, J.L.C. and HUANG, T.S. (1983), 'Unique reconstruction of a band-limited multidimensional signal from its phase or magnitude', *Journal of the Optical Society of America*, Vol. 73, No. 11, November, pp. 1446–1450.

SANZ, J.L.C. and HUANG, T.S. (1985), 'Polynomial system of equations and its applications to the study of the effect of noise on multidimensional Fourier transform phase retrieval from magnitude', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, pp. 997–1004.

SANZ, J.L.C., HUANG, T.S. and CUKIERMAN, F. (1983), 'Stability of unique Fourier-transform phase reconstruction', *Journal of the Optical Society of America*, Vol. 73, No. 11, November, pp. 1442–1445.

SASAKI, O. and YAMAGAMI, T. (1987), 'Phase-retrieval algorithms for nonnegative and finite-extent objects', *Journal of the Optical Society of America A*, Vol. 4, No. 4, April, pp. 720–726.

SAXTON, W.O. (1978), *Computer Techniques for Image Processing in Electron Microscopy*, Advances in Electronics and Electron Physics, Supplement 10, Academic Press, New York.

SECHEN, C. (1988), *VLSI Placement and Global Routing Using Simulated Annealing*, Kluwer Academic Publishers, Boston.

SELDIN, J.H. and FIENUP, J.R. (1989), 'Ayers/Dainty blind deconvolution applied to phase retrieval', Submitted to: *Journal of the Optical Society of America A*.

SEZAN, M.I. and STARK, H. (1982), 'Image restoration by the method of convex projections: Part 2 - Applications and numerical results', *IEEE Transactions on Medical Imaging*, Vol. MI-1, No. 2, October, pp. 95–101.

SEZAN, M.I. and STARK, H. (1983), 'Image restoration by convex projections in the presence of noise', *Applied Optics*, Vol. 22, No. 18, 15th September, pp. 2781–2789.

SHI, Y.Q. and BOSE, N.K. (1988), 'Some results in nonnegativity constrained spectral factorization', *Optics Communications*, Vol. 68, No. 4, 15th October, pp. 251–256.

SHORE, J.E. and JOHNSON, R.W. (1980), 'Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy', *IEEE Transactions on Information Theory*, Vol. IT-26, No. 1, January, pp. 26–37.

SILVER, S. (Ed.) (1965), *Microwave Antenna Theory and Design*, Dover Publications, New York.

SINTON, A.M. (1986), *Contributions to Astronomical and Medical Information Processing*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand, May.

SLEPIAN, D. (1976), 'On bandwidth', *IEEE Proceedings*, Vol. 64, March, pp. 292–300.

SMITH, W.E., BARRETT, H.H. and PAXMAN, R.G. (1983), 'Reconstruction of objects from coded images by simulated annealing', *Optics Letters*, Vol. 8, No. 4, April, pp. 199–201.

SMITH, W.E., PAXMAN, R.G. and BARRETT, H.H. (1985), 'Image reconstruction from coded data: I. Reconstruction algorithms and experimental results', *Journal of the Optical Society of America A*, Vol. 2, No. 4, April, pp. 491–500.

SONDHI, M.M. (1972), 'Image restoration: the removal of spatially invariant degradations', *IEEE Proceedings*, Vol. 60, July, pp. 842–853.

STOCKHAM, JR., T.G., CANNON, T.M. and INGEBRETSON, R.B. (1975), 'Blind deconvolution through digital signal processing', *IEEE Proceedings*, Vol. 63, No. 4, April, pp. 678–692.

TRUSSELL, H.J. and CIVANLAR, M.R. (1984), 'The feasible solution in signal restoration', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 32, No. 2, April, pp. 201–212.

WALKER, J.G. (1981), 'The phase retrieval problem. a solution based on zero location by exponential apodisation', *Optica Acta*, Vol. 28, No. 6, June, pp. 735–738.

WALKER, J.G. (1982), 'Computer simulation of a method for object reconstruction from stellar speckle interferometry data', *Applied Optics*, Vol. 21, No. 17, pp. 3132–3137.

WALTHER, A. (1963), 'The question of phase retrieval in optics', *Optica Acta*, Vol. 11, pp. 41–49.

WOOLFSON, M.M. (1987), 'Direct methods - from birth to maturity', *Acta Crystallographica*, Vol. A43, No. 5, September, pp. 593–612.

YOULA, D.C. (1978), 'Generalised image restoration by the method of alternating orthogonal projections', *IEEE Transactions on Circuits and Systems*, Vol. CAS-25, pp. 694–702.

YOULA, D.C. (1987), 'Mathematical theory of image restoration by the method of convex projections', in STARK, H. (Ed.), *Image Recovery : Theory and Application*, Academic Press, Orlando, Florida, Chap. 2, pp. 29–77.

YOULA, D.C. and WEBB, H. (1982), 'Image restoration by the method of convex projections: Part 1 - Theory', *IEEE Transactions on Medical Imaging*, Vol. MI-1, No. 2, October, pp. 81–94.